Connectionist Signal Processing System
Characterization of representation in neural netwo

Deepak Murthy

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

March 1996

Connectionist Signal Processing Systems:
Characterization of representation in neural networks

A Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

by

Deepak Murthy

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

March 1996

CERTIFICATE

It is certified that the work contained in the thesis entitled *Connectionist Signal Processing Systems: Characterization of representation in neural networks*, by DEEPAK MURTHY, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

PRKRao

PROF. P R K RAO
Department of Electrical Engineering
Indian Institute of Technology Kanpur

March 1996

Synopsis

Name of Student: Deepak Murthy Roll No : 8910463

Degree for which submitted: Ph D Department· Electrical Engineering

Thesis Title: **Connectionist Signal Processing Systems: Characterization of representation in neural networks**

Name of thesis supervisor· Prof. P R K Rao

Month and year of thesis submission: March 1996

Conventional signal processing based on a parametric representation of signal spaces and, as a consequence, operations involving either an estimation of the parameter(s) given sufficiently many observations or a manipulation of the parameter(s) to obtain the desired signals needs an exhaustive account of the dependence of the output signal on the input signal in terms of criteria with global scope, if not in terms of closed form expressions. Alternative approaches, generally non-linear, based on grammatical formulations of signal spaces and operations have also been suggested in the literature to overcome some of the limitations attributed to linearity in conventional signal processing.

Several instances of signal processing, generally involving a *subjective* element in the processor–though not devoid of invariance, *eg*, recognition of hand written characters, facial recognition, texture identification *etc*, exist wherein neither a parameterization nor a grammatical formulation of signal spaces is feasible due to insufficient understanding of the processes underlying the signal generation and/or the volume of data being grossly inadequate to establish input-output relationships. However, such situations can be described through finitely many prototype inputs for which the outputs are known either completely or partially. Processors of this kind have been realized through a class of hierarchical non-linear dynamical systems termed artificial neural networks wherein the processor belongs to a parametrically described space and the objective is to estimate (learn) processor parameter(s) given examples of input-output association.

Current neural network research exhibits a plethora of networks, each concentrating on representing specific types of input-output relationships with accompanying procedures to estimate the processor parameters given the examples of association. However, the issues of representation have not been adequately investigated with the consequence that no satisfactory criterion exists whereby one can decide on the architecture of the network necessary for a given situation of processor realization. In addition, little attempt is made at providing an axiomatic framework in which neural networks architectures and processor realization can be discussed

Overview of the Thesis

I focus my investigations on four key issues related to the representation of signal processors with neural networks, each not unconnected with the others. Representation is interpreted, in this thesis, as a decomposition and/or synthesis of the desired function through 'basis' functions that are not chosen *a priori*, but are synthesized to suit the requirements of processor realization: the requirements are specified through finitely many 'examples' of the desired association. The 'basis' functions are, however, not restricted to be dependent on the family of functions under consideration.

The investigation is initiated by considering representation in isolated neurons from the perspective of preservance of input spaces in input-output associations: preservance of mappings is defined in terms of one-one correspondence, order preservation and preservation of regularity. I establish the existence of weights, corresponding to isolated neurons, that accommodate a preservance of the collection of binary vectors in an Euclidean space and identify the class of preservance weights corresponding to the collection of binary vectors. Though this class has uncountably many elements, these elements are organized in finitely many 'directions': the number of directions are related exponentially to the dimensionality of the collection of binary vectors.

Preservance is shown to extend to enlarged discrete spaces derived as certain finite unions of scaled and translated versions of the collection of binary vectors, however, without an alteration in the class of preservance weights. Functions on such discrete spaces under preservance are equivalent to sequences on the input space and as a consequence linear separability is characterized in terms of the number of sign-transitions in the sequences and learning is shown to be equivalent to an enumeration of weights in the class of preservance weights and a search for threshold in a linearly ordered space.

The radix of numbering is shown to have little influence on preservance though the cardinality of the discrete space preserved increases with the radix and the preservance weights tend to bunch around the coordinate axes. Corresponding to every non-null weight vector in an Euclidean space a preservance input space, defined as a discrete space for which this weight would be a preservance weight, is identified and the preceding discussion is shown to extend to such an input space, though with appropriate rotations of the relevant coordinate frames.

Representation in layered neural signal processors is the next issue considered in this thesis, however, the investigation is restricted to the specific case of feed-forward ensembles realizing maps on preservance input spaces as linear combinations of neural responses. A single layer processor restricted to have identical weights in all nodes is first considered: the thresholds are, in contrast, allowed to be distinct. Such a structure is shown to represent all dichotomies on a preservance input space whose preservance weight is used as the common weight.

The number of nodes is no more than the number of distinct level-transitions in the sequences along the preservance weight that represent functions over the preservance input space and the number of level-transitions is used to test minimality of an architecture. Learning in this processing structure is shown to involve a process of approximating the collection of inputs described in a training set by a preservance input space, a search for a threshold in a linearly ordered space and an analytical solution for the coefficients of linear combination. A similar situation is shown to exist when the weights in the constituent neurons are distinct preservance weights of the same preservance input space.

Multi-layered neural signal processors modeled to realize functions as linear combinations of neural network responses are shown to be functionally equivalent to single layer neural signal processors, however, fewer nodes are needed to represent a given function when compared with that necessary in the corresponding minimal single layer processor. This is true only when the number of layers is smaller than the number of nodes needed in the minimal single-layer processor. As the preservance input space is discrete, an algebraic characterization of function realization with neural networks is considered to establish that linearly separable dichotomies are exactly those partitions on (semi) lattices wherein each member, of the partition, is a semi-lattice.

Issues of representation in neural signal processing architectures form the third topic of investigation in this thesis. Typed neural signal processors are defined on continuous spaces, the type number reflecting the degree of layering. Functions realized by neural signal processors of all types are shown to be dense in the space of continuous functions: this is an extension of similar results established, on equivalents of type-1 processors, by **Cybenko** (1989) and **Hornik, Stinchcombe & White** (1989), to cascades of type-1 processors. Through a study of the functional nature of neural signal processors, four axioms are suggested to describe the current architectural commitments in neural signal processing activity: these axioms are sufficiently general to aid a unified study of neural signal processing architectures.

1. Axiom of Organization A neural signal processor is composed of (layers of) three operational stages: measurement, discrimination and aggregation in that order. Preprocessing, if any, (preceding, or incorporated in, the measurement) is sought to be represented in a neural basis. Measurements are effected on an observation space constructed as the Cartesian product of the input space and a relevant subspace of a union of the space of responses of the distinct layers.

2. Axiom of Measurement. A neural signal processor, through the measurement functions in each of the processing (decision making) nodes, induces a foliation, of codimension at least one, in the input manifold. This foliation forms the basis of synthesizing (approximating) the desired level curves of the function.

3. Axiom of Discrimination. A neural signal processor, through its discriminatory functions, renews the foliations, induced on the input space by the measurement functions, through a transformation, of the stems of the foliations, with at least one of the following properties.

   (a) alter the indexing of leaves to retain distinctness in a finite non-zero number of local regions of the input space,

   (b) introduce multiple components in the leaves,

   (c) associate, to at least one component of a leaf of the foliation due to discrimination, uncountably many leaves of the foliation due to measurement.

   Re-foliations provide the basis for establishing equivalences between members (elements) of the input space in ways not possible through the chosen measurement functions.

4. Axiom of Aggregation. A neural signal processor, through its aggregation function, synthesizes (or approximates) the level regions of processor response through a foliation on the Cartesian product of the stems of foliations on the input space due to discrimination. Concepts, in neural signal processors, are identified with the level regions of processor response.

These axioms, coupled with the earlier stated algebraic characterization of linear separability, suggest that the 'paradigm' of neural computing, (specifically the notions of 'learning' and 'generalization') is not restricted to processors effecting maps between (vector) spaces of numbers. As the notion of a foliation (**Lawson**, 1974) is one of inducing a partition on a space such that the members of the partition belong to an indexed collection, these axioms allow attention to be directed towards a unified treatment to neural computing, especially the analysis (and synthesis) of representation with neural networks. In particular, these axioms provide a framework wherein a formulation of problems related to decidability, solvability and completeness that dominate the theory of computing–these problems lead to queries about the capabilty of the neural computing paradigm to address issues related to the design of neural networks through the paradigm of neural computing–and a relative evaluation of the formalism of Turing Machines with the paradigm of Neural Networks can be attempted. A unification, however, is not in the scope of this thesis.

At an operational level neural signal processors effect (point-wise) nonlinear transformation between integral transforms: this interpretation allows representation in neural networks to be contrasted with other approaches to signal processor realization. The resulting constituents are used to suggest an interpretation to a function representation theorem due to **Kolmogorov** (1957a): this interpretation is different from that provided by **Hecht-Nielsen** (1987a), **Kůrková** (1992) and **Kovačec & Ribeiro** (1993). Learning, under this interpretation, is equivalent to kernel design. The possibility of a solution to learning with *a priori*, but partial, knowledge of weights, a situation relevant in hybrid networks, is indicated by incorporating neural network based function realization for the kernels of the integral transforms.

Localization in the representation of neural signal processors is the final issue considered in this thesis. A localization in representation is shown to result from an influence of the kernels of integral transforms as well as from the mechanism of (point-wise) association between integral transforms. Localization resulting from kernels is shown to restrict the choice of weights in individual neurons to the linear span of window functions (sequences), however, there is no restriction on the constituent window functions (sequences). I also establish that the mechanism of association is restricted to have all derivatives (those that exist) in the linear span of window functions, effectively suggesting that in the connectionist approach to signal processor realization, signals and their processors are both capable of being described in comparable, possibly same, 'basis' space: this feature would be helpful in a formulation of neural network based systems which decide on the processing characteristics of neural networks.

A characterization of localization in terms of wavelet transforms is considered to suggest the operational sense of 'basis' function synthesis in neural network representations. This characterization is different from that provided by **Zhang & Benveniste** (1992) and **Pati & Krishnaprasad** (1993). Concepts represented in neural signal processors are shown to reflect evaluation of intra-pattern and inter-pattern features, the former is influenced by localization due to measurement and aggregation kernels and the latter is a consequence of the mechanism of association between the integral transforms of measurement and aggregation. I also establish that localization in the intra-pattern and inter-pattern predicates restricts concepts represented by every node in a neural signal processor to a localized region, with one or more components, in the sheaf of input patterns.

I have considered kernels of the reproducing type as a specific example of localization in the integral transforms of measurement and aggregation. These reproducing kernels have been shown to extend the notion of preservance–defined earlier on discrete input spaces–to input spaces that are continuous, however, with the limitation that not all reproducing kernels are representative of preservance weights, and, in the same way, not all kernels representing preservance weights exhibit the reproducing property.

Based on the discussion of **Nashed & Walter** (1991) that every reproducing kernel is associated with a sampling theorem, I have established that the nature of representation in neural signal processors is in the sense of approximating concepts that are defined on continuous domains through finite number of (non-uniformly) spaced samples: the finiteness of the number of samples is assured when the concepts are of a localized nature and non-uniformity in sampling is admitted by the Paley-Wiener sampling theorem (*op cit*). This result implies that conventional neural networks–*ie*, networks of finitely many neurons, each with finitely many inputs–represent concepts in a continuum if the kernels are of the reproducing type.

An attempt at representing the (reproducing) kernels of the integral transforms of measurement and aggregation via the paradigm of neural signal processing suggests that in the earlier stated notion of representation, the 'basis' functions synthesized are related to members of a (wavelet) frame. The characteristics of the basic wavelets in the frame are decided by the degree of layering incorporated in the neural networks that synthesize the kernels of the measurement and aggregation integral transforms: larger the number of layers, greater is the degree of localization effected by the basic wavelets.

Based on the characterization of representation in neural networks presented in this thesis, I have conjectured that the nature of representation in multi-layered networks is of the following kind: 'shallow' networks are well suited for representing processors that have formal descriptions (*ie*, a description involving rules of association) whereas 'deep' networks are necessary when the entities operating in a formal system needs to be identified/discovered. In other words, 'shallow' networks are good in symbol processing while 'deep' networks are necessary for 'symbol synthesis.' Present neuro-anatomical evidence does not seem to refute this conjecture in that the cortex and neo-cortex, the seat of (conscious) symbolic activity, is organized to have few layers, each with a wide spread of interconnections. In contrast, the mid-brain, whose functionality is not known in sufficient detail, but is believed to be responsible for (sub conscious) associations (part of which is the long term memory trace) are 'deep' networks with localized connections.

Organization of the Thesis

The findings of my investigation together with a review of signal processing with neural networks is organized as a report consisting of seven chapters. An introduction to the idea of automated information processing, stressing on the connectionist approach to signal processing is presented in the first chapter. Some of the historical aspects of the connectionist approach to information processing are also incorporated. This chapter also dwells on the motivations for the present investigation and, as a preface, presents an overview of the thesis accompanied by an outline of the thesis organization.

A review of signal processing with neural networks is presented in Chapter 2. The notion of signals, their processing and associated abstractions followed by prominent models describing the processing in isolated neurons and neuronal ensembles are briefly introduced to provide the relevant background, terminology and notations. An outline of the approaches available in the literature for the realization of signal processors through neural networks is also incorporated. In addition, the notions of intelligence and information processing are cursorily reviewed in an appendix to supplement the contents of the first two chapters.

The issue of representing signal processors in isolated neurons is taken up in Chapter 3. In this chapter, I introduce the notion of preservance and establish the existence of preservance weights. Preservance is initially established on the collection of binary vectors in a Euclidean space of dimensionality n and extended to discrete spaces constructed from the collection of binary vectors through scaling and translation. A characterization of the discrete input space accommodating preservance, the collection of weights that form preservance weights and functions represented on such spaces are incorporated. This chapter ends with a discussion on the extension of preservance to discrete spaces identified with numbering systems of a radix other than binary and a construction of preservance input spaces corresponding to arbitrary, but non-null, weights.

Neural signal processor realization in layered ensembles of neurons is focused in Chapter 4. The influence of preservance on function realization in single layered neural signal processors is taken up first and this study is utilized in the study of function realization in multi-layered neural signal processors. An identification of preservance input spaces appropriate to the collection of inputs described in a training set and the attendant issues in the representation of input spaces is considered in this chapter. The algebraic characterization of representation in neural signal processors on discrete input spaces forms the final component of this chapter.

Characterization of neural signal processing architectures forms the theme of Chapter 5. In this chapter, I introduce neural signal processors with types and consider the potential for representation in neural signal processors: the processors are considered operating on continuous input spaces. The functional characteristics of neural signal processors, axioms of neural signal processing and the suggestion for an operational paradigm of neural signal processing are considered in this chapter. A study of representation in neural signal processors in terms of function approximation is the final topic in this chapter.

In Chapter 6, the issue of localization in the functions represented by neural signal processors on continuous input spaces is investigated. The nature of localization is first studied in the case of isolated neurons and then the study is carried over to feed-forward layered ensembles of neurons. Characterization of localization in terms familiar in the literature of signal processing and implications of localization on the nature of processing are considered in this chapter. The 'basis' functions through which signal processors are realized in neural networks are related to wavelet transforms. In Chapter 7, I summarize the findings of my investigation and suggest directions of further study.

References

Cybenko, *George* (1989) Approximation by superpositions of sigmoidal functions *Mathematics of Control, Signals, and Systems*, 2·303–314.

Hecht-Nielsen, *Robert* (1987a). Counterpropagation networks. *Applied Optics*, 26 4979–4985. Also presented in **Hecht-Nielsen**, 1987b

Hecht-Nielsen, *Robert* (1987b). Counterpropagation networks, in *Proceedings of the First International Conference on Neural Networks, Vol II* (Edited by **Caudill**, *Maureen* and **Butler**, *Charles*), pages 113–12, 19–32. Also presented in **Hecht-Nielsen**, 1987a.

Hornik, *K*, **Stinchcombe**, *Maxwell* and **White**, *Halbert* (1989) Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5).359–366.

Kolmogorov, *A N* (1957a). On the representation of functions of several variable by superpositions of functions of one variable and addition. *Translations of the American Mathematical Society*. Translated from the Russian original **Kolmogorov** (1957b)

Kolmogorov, *A N* (1957b). On the representation of functions of several variable by superpositions of functions of one variable. *Dokl. Nauk.* In Russian.

Kovačec, *Alexander* and **Ribeiro**, *Bernardete* (1993). Kolmogorov's theorem: From algebraic equations and nomography to neural networks, pages 40–47.

Kůrková, *Věra* (1992). Kolmogorov's theorem and multilayer neural networks. *Neural Networks*, 5.501–506.

Lawson, Jr, *H Blaine* (May 1974). Foliations *Bulletin of the American Mathematical Society*, **80**(3):369–418.

Nashed, *M Zuhair* and **Walter**, *Gilbert G* (1991). General sampling theorems for functions in reproducing kernel Hilbert spaces. *Mathematics of Control, Signals, and Systems*, 4:363–390.

Pati, *Y C* and **Krishnaprasad**, *P S* (January 1993). Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations. *IEEE Transactions on Neural Networks*, 4(1):73–85.

Zhang, *Qinghua* and **Benveniste**, *Albert* (November 1992). Wavelet networks. *IEEE Transactions on Neural Networks*, 3(6):889–898.

[Marco Polo:] 'Sire, now I have told you about all the cities I know.'

[Kublai Khan:] 'There is still one of which you never speak.'

Marco Polo bowed his head.

'Venice,' the Khan said.

Marco smiled 'What else do you believe I have been talking to you about?'

The emperor did not turn a hair. 'And yet I have never heard you mention that name.'

And Polo said: 'Every time I describe a city I am saying something about Venice '

'When I ask about other cities, I want to hear about them. And about Venice, when I ask you about Venice.'

'To distinguish the other cities' qualities, I must speak of a first city that remains implicit. For me it is Venice.'

'You should then begin each tale of your travels from the departure, describing Venice as it is, all of it, not omitting anything you remember of it.'

. .

'Memory's images, once they are fixed in words, are erased,' Polo said. 'Perhaps I am afraid of losing Venice all at once, if I speak of it. Or perhaps, speaking of other cities, I have already lost it, little by little.'

— Italo Calvino
in *Invisible Cities*,
Picador, London, 1979.

Acknowledgments

Research, howsoever trivial, irrelevant, or insignificant, is not commonly the sole effort of a single person, and even the slightest semblance of courteousness demands an acknowledgment of help and guidance received, and it is customary not to mention the hindrances and hurdles faced during this intellectual tour in the province of ideas I humbly submit that the investigations presented herein have been possible only because of the invaluable and timely advice and assistance generously provided by several people. Fully accepting an established tradition I acknowledge, with heartfelt pleasure, all the assistance I have received

I thank my parents, Er. BSN Murthy & Vatsala Murthy (my first educators), sister, Dr. Mamata Murthy, and other relatives for having provided me the necessary support, many a times sacrificing their own personal needs and expectations. Without their support, I would not have been able to withstand the hardships, especially in the past few years of my programme. In particular, I acknowledge the preparation I received, by nature as well as nurture, from Late Er. LN Jois and Late Prof. BN Shamarao, my grandfathers, in addition to my parents, to persevere in an intellectual journey with an unebbing curiosity and fortitude

I would be failing in my duty not to acknowledge Prof. PRK Rao who inspired me to seek not only the technical aspects of signal processing and neural networks, but also issues pertaining to philosophy, science, technology, and their appreciation I am extremely grateful to him for having allowed me to continue my investigations despite a denial of basic facilities and financial assistance and for encouraging me to work without the distraction of visiting home, though, in a warped system of values, the restraints I faced are considered disadvantageous, monetarily and emotionally

I also express my gratitude for his having allowed me to persist in my investigations without the embarrassing vexation of paper publications and conferencing. A summer school on neural networks (conducted, in March 1990, by the Center for Theoretical Studies, Indian Institute of Science Bangalore) that I attended on his recommendation, followed by his course on neural networks, helped me acquire the relevant background on neural networks reasonably painlessly. I thank Prof Chandan DasGupta for allowing me to participate in this summer school

During my prolonged stay at IIT Kanpur, I have had the opportunity to interact with several members of the faculty from different departments, and some of these interactions have given me sufficient inspiration in assuaging my curiosities In particular, Prof. VP Sinha, Prof. PK Ghosh, Prof. PC Das, and Prof. G Barua besides my thesis supervisor, have inspired me in the areas of their professional expertise and eminence, allowing me to indulge in fantasies of a possible unification of different domains of inquiry

I also thank all friends and acquaintances, who have cherished, without complaint, my ambition for learning, and have helped me acquire the confidence and strength of walking unaccompanied, even in an arduous journey Several of my fellow students have enriched my experiences and enlivened my stay by willingly participating in numerous discussions, many of these extending into the wee hours of the night, or long walks within the campus, focusing on several issues relevant to our times

Most notorious among them are Venkatesh, Sundar Rajan (Siva), Jitendra Das, Praveen Bhatia and Tarakeshwar I can never forget my fellow residents in Hall IV, in particular, Vinod Kumar, Jawed, S K Verma, Atul Verma, Alok Sharan and G K Singh who through their boisterousness, levity, and kindred spirit, lessened the ordeal of research. Last but not least, I have to thank the monkeys and magnificent peacocks cohabiting the campus, who, unmindful of the present clamor of the 'rights' to 'intellectual property,' have been generous enough to enrich the members of this campus community with their antics and pride, thereby animating an otherwise monotonous academic environment.

IIT Kanpur

Deepak Murthy

Dedicated to
my Parents and Sister
with admiration and affection.

Contents

List of Tables

List of Figures

Chapter 1

Connectionist Jnformation Processing: An introduction

It is a degradation to a human being to chain him to an oar and use him as a source of power; but it is an almost equal degradation to assign him purely repetitive tasks in a factory [or institution], which demand less than a millionth of his brain power. But it is simpler to organize a factory or galley which uses individual human beings for a trivial fraction of their worth than it is to provide a world in which they can grow to their full stature.

—Norbert Wiener
in *The Human Use of Human Beings – Cybernetics and Society*,
Houghton Mifflin Company, Boston, 1950.

In our present sustained concern for 'progress' and 'development' the associated incessant increase in demand for handling of 'materials' could not have been met at the present levels of success without recourse to automation. 'Materials,' in this discussion, need to be considered in a sense more abstract than our common interpretation of material being merely a morphological manifestation of matter.[1] Avoiding detail, it is sufficient to note that the abstract notion of materials includes isolated or combined participation of manifestations of matter, energy and information: this abstraction regarding materials and material handling is supported by the views of **Diebold** (1952)[2] and **Stonier** (1990). This thesis is restricted to the 'information dimension' of materials.

Handicrafts, industrially fabricated products, energy in its various forms, human speech, music, images, computer programs and even mental states, viewed as states of the brain may be considered as examples of the *abstract material.* (See **Churchland**, 1986; **Churchland & Sejnowski**, 1994, for a discussion on the materialist reduction of mental events to events in the brain.) Automation is a historical process involving two key aspects.

(a) An identification of certain human endeavors which, though routine, are considered essential for human survival and well being.

[1] The necessity for considering the notion of material and its handling, at this abstract level arises in the context of the importance we have assigned, in our daily lives, to each of the concrete examples of the abstraction.

[2] References are cited through the last name(s) of the author(s) (or editor(s)) and the year of publication of the manuscript. A list of references has been included at the end.

(b) An identification of processes/mechanisms by which routine human activities can be mimicked anthropomorphically at least at the functional, if not at the phenomenological, level: these machines (mechanisms) are expected to compete with human presence in routine material handling and even guide the genesis and ontology of [machine specific] metaphors (models), ultimately challenging human existence to conform to these [new] metaphors.

Material handling, by automatic means, necessitates three basic component mechanisms: sensors, effectors and control (or coordination). *Sensors* are needed to detect (or measure) physical occurrences (location and extent) and the acceptability of processing to which the material has been subjected to. The desired steps of processing on the material (as simple as translations, in space and/or time) are implemented through *effectors*. *Control* (*coordination*) ensures that material flow through all sections of the automated 'plant' is unencumbered and the product quality is assured at every stage of processing. In the most popular presentations of automated material handling, the three basic components are likened to human organs, *viz*, sensory organs, motor organs (chiefly limbs and fingers) and the brain respectively.

Information processing, especially decision making, is one of the crucial aspects of the control (coordination) mechanisms. The complexity of information handling, *ie*, acquisition of information from sensors, transmitting information between sensors and controllers as well as controllers and effectors, implementation and/or execution of commands by effectors and decisions to be taken by the controller, is increasing day by day with escalating demands on 'productivity' and increasing sophistication in the technology of the material processing steps. Incorporation of the abilities acquired in the automation of material handling to the processing of information is an immediate and natural, reaction to overcome the hurdle of information processing complexity.

The automation of information processing generates the hope and desire, that *intelligence*, which for long has been regarded as a natural privilege of human beings, is expressed through machines and that this *automated intelligence* (normally termed as *artificial intelligence*) will be available for the control and coordination of automated systems (including those involved in the automation of information processing). Several approaches have been suggested for the mechanized expression of intelligence, however, these can be broadly categorized under two distinct labels: top-down and bottom-up approaches, typical representatives being *symbolic artificial intelligence*, or *Classical AI* and *neural networks*, or *Connectionist AI* respectively.[3] A preliminary discussion on the nature of automated intelligence and a cursory review of the approaches to artificial intelligence have been included in Appendix A.

[3] These approaches have, ever since their inception, been dogged by controversies; some arising out of a [furious] debate between the various schools of thought regarding the nature of human intelligence and the approach to automated intelligence. (See **Olazaran**, 1993, for a discussion of the sociological history of the controversies in the approaches to artificial intelligence)

In this chapter, as a preface to the thesis, I will outline the connectionist approach to artificial intelligence in order to highlight the (important) operational concerns and technical issues that have animated the understanding of intelligence. As neural networks are increasingly finding acceptance, in the signal processing community, the motivations for a study of connectionist information processing systems, in particular, issues related to representation of signal processors with neural networks, followed by statement of the problems addressed and a preview of the thesis have been included. An overview of the organization of the thesis has been provided at the end of this chapter to facilitate an easier movement through the contents

1.1 Connectionist Artificial Intelligence

Intelligence having been viewed as a consequence of information processing, one of the central issues of intelligence concerns information characterization and the nature of information representation.[4] The central questions of automated intelligence, viewed in the connectionist perspective enunciated by **Rosenblatt** (1958), are:

[4] The view that intelligence is a consequence of information processing, while being dominant, is not, however, shared by all interested in the automation of intelligence. A criticism of information processing models in perceptual categorization and generalization, the key aspects of an expression of intelligence, is offered by **Edelman** (1987). This criticism brings one of the important limitations of information processing models to intelligence, *viz*, the *Homunculus* problem.

i. How is information about the physical world sensed, or detected, by the biological system?

ii. In what form is information stored, or remembered?

iii. How does information contained in storage, or in memory, influence recognition and behavior?

Perceptrons and neural networks inspired by biological information processing, in particular, the architecture of the brain, are claimed to address these issues. The central theme of the connectionist paradigm, despite the varied interpretations, is that

> Whatever information is retained must somehow be stoied as a *preference for a partıcular response*; *ie*, the information is contained in *connections* or *associations* rather than topographic representations. (The term *response* . . . should be understood to mean any distinguishable state of the organism, which may or may not involve externally detectable muscular activity [*ie*, *state* in the language of dynamical systems].)

This view, expressed by **Rosenblatt**, 1958, is supported by other investigators. (See Appendix A.)

Artificial neural networks, or neuromorphic systems, do not have a standardized definition, or terminology.[5] However, the following def-

[5] This is partly because of the differing interests in the community of investigators charmed into a study of neural networks.

inition by **Sage & Withers** (1990) based on the definition suggested in the DARPA Neural Network study captures the consensus in the existing definitions.[6]

> A system composed of many simple processors, fully or sparsely connected, whose function is determined by the connection topology and strengths
>
> This system is capable of a high level function such as adaptation or learning with or without supervision as well as lower level functions such as vision and speech preprocessing.
>
> The function of the simple processors and the structure of the connections are inspired by the study of biological nervous systems.

Hecht-Nielsen (1990) suggests a more technically elaborate definition for neural networks and **Fiesler** (1994) has proposed a standardization in the terminology of neural networks

Before discussing the operational history of neural networks, it will not be out of place to have a brief digression to understand the scope of

[6] A moment's indulgence in the luxury of abstraction would reveal that the common refrain in all of the existing definitions is that neural networks are *function fields over partially ordered index spaces*: as of yet, however, the collection of functions are indexed over lattice-points In an abstraction of this form, neural networks share the same universe as (Universal) Turing Machines, Finite State Machines, Grammars, Normal Algorithms, *etc.* With this abstraction, it is important to seek out the interplay between inter-function interactions and the macroscopic functional specificities (or properties), especially to understand the nature of cognition, *ie*, automated intelligence, that can be accounted for by models sharing the above abstraction. Neural networks formulated as function fields over partially ordered lattices will provide a framework well suited for a study of the representational characteristics of universal neural networks.

interpretation available with the term *Artificial Neural Networks.* On the basis of an analysis of meaning, we can argue that four (subtly) distinct, yet interacting, activities are valid [operational] candidates under the common banner of artificial neural networks and each of these is important in the automation of intelligence. These altering interpretations occur due to minor variations from the DARPA definition.

The first of these interpretations stems from *our* common understanding of the adjectives *artificial and neural*, wherein the discussion is of networks of [processing] elements each of which is a mimicry of isolated real world neurons, the specific characteristics of the processing elements and of the interconnections between the processing elements retaining empirically established properties. In this activity, the focus is one of establishing models for isolated real world neurons and to meticulously study the various kinds of interconnections exhibited in biologically expressed neuronal ensembles – typically, brain – and to relate the observed structures to biological functions. Such a study is to be expected, commonly, in established departments of neurobiology (incorporating neuroanatomy and neurophysiology).

An activity encompassing a study of methods by which to cultivate networks of neural-like processing elements, possibly as replacements of existing brains, aided by the interpretation of *artificial neurons* in the sense of synthesized neurons is the second valid interpretation of the term *artificial neural networks*: it is not difficult to notice that such an activity can sustain only on the knowledge generated by activities falling in the first category. This activity, essentially being a study of biologically compatible devices aiding (human) intervention in the otherwise (mal)functioning systems, is to be anticipated under heads like bio-engineering, bio-electronics, *etc.* Here, it is important to note that the stress is on getting the synthetic neurons to act as reasonable substitutes (or surrogates) to real life neurons.

Interest in bio-engineering promotes an activity wherein cultures of neural ensembles are interfaced (electronically) to engineering systems and motivates inquiries into the computational advantages offered, over conventional electronics, by biological information processing substrates. This activity, foreseeable under heads like *neuro-technology*, *bio-informatics*, *etc*, and a third valid interpretation of *artificial neural networks*, relies on the knowledge provided by *neuro-science* to cater to specific processing requirements. In such an activity, the focus, typically, is on realizing the desired information processing requirement given the characteristics of the (biological) processing substrate and in a sense, is n●t very different from activities in conventional electronics.

While the first category of activity underlines the possibility of *neuro-science* and the second and third categories anticipate *neuro-[bio]-technology*, we cannot surely miss out on another valid interpretation allowing for an activity supporting a study of possible alternatives to the characteristics of neurons (processing elements) and inter-neural interconnections. This interpretation, driven by issues of *practical* realizability, seeks out an exploration of possible structures and thereby attempts an understanding of the dimension of automated intelligence.

Indeed, this approach, which may be termed as *neuro-engineering*, is the one activity that has been the fancy of many an investigator in the present time: a fancy, not necessarily in the sense of an irrational choice in the presence of other viable information processing possibilities, however, in view of the fact that several information processing situations are being handled by methods involving neural networks, *ie*, neural networks are being viewed as a *panacea* for all information processing situations. This thesis too will be confined in its attention to the neuro-engineering aspect of operational interpretation. It should however, be noted that neuro-science and neuro-philosophy (discussed mainly in **Churchland**, 1986) are not unimportant for an understanding of neuro-engineering.

Research activity in neural networks has, since its beginning in the work of **McCulloch & Pitts** (1943), taken on all the above interpretations, in particular those provided by the first and last categories. Perceptrons were the first non-trivial neural networks to be investigated. The focus in neural networks has always been one of information representation and one of the important manifestations of this focus, in addition to that of architectural types, is in the 'automatic' selection or search of (tunable) connection strengths between processing nodes. Biological inspirations have guided the labeling of this activity as *learning*, or *training*. An equally important consequence of the representational focus is in *generalization*, *ie*, the problem of extending the scope of the knowledge base, represented through learning, to input situations not contained in the training repertoire: this capability is projected as one of the strong points of the sub-symbolic paradigm over the symbolic representational framework

Initial investigations in perceptrons were limited to a single level of adaptive weights and the discovery of a training algorithm triggered intense investigations. However, this was to be short-lived as automated training of multi-layered networks turned out to be elusive. In this context, **Minsky & Papert** (1969) established some serious limitations:

> No diameter-limited perceptron [*ie*, a perceptron wherein each constituent processing node evaluates a local predicate] can determine whether or not all the parts of any geometrical figure [incident on the retina] are connected to one another! ..
>
> Part of the attraction of the perceptron lies in the possibility of using very simple physical devices – "analogue computers" – to evaluate the linear threshold functions. It is perhaps generally appreciated that the utility of this scheme is limited by the sparseness of *linear* threshold functions in the set of *all* logical functions. However, almost no attention has been paid to the possibility that the set of linear functions which are *practically* realizable may be rarer still ...

> The perceptron has shown itself worthy of study despite (and even because of!) its severe limitations. It has many features to attract attention: its linearity; its intriguing learning theorem; its clear paradigmatic simplicity as a kind of parallel computation. There is no reason to suppose that any of these virtues carry over to the many-layered version. Nevertheless, we consider it to be an important research problem to elucidate (or reject) our intuitive judgment that the extension is sterile. Perhaps some powerful convergence theorem will be discovered, or some profound reason for the failure to produce an interesting "learning theorem" for the multi layered machine will be found.[7]

In addition, Minsky and Papert established that simple predicates like that of parity (and connectedness) were not represented by perceptrons and pointed out the futility of the usage of perceptrons in view of the limited set of representable predicates.

These observations motivated several investigators to opt for information processing approaches different from perceptrons and quite a large number got interested in the symbolic processing paradigm. Quite independent of investigations in perceptrons, **Widrow & Hoff** (1960)–see also **Widrow & Winter** (1988)–had investigated the suitability

[7] It is noteworthy to remark that in the same year that Minsky and Papert came out with their book, **Bryson & Ho** (1969) suggested, in the field of optimal control, an algorithm for automatic specification of parameters in multi-stage controllers. This algorithm shares many features with other (stochastic) gradient search algorithms like Delta rule and Error Back-Propagation.

of perceptron like ADALINES (adaptive linear elements) and MADALINES (many ADALINES) in signal processing applications. In these investigations, a variant of stochastic gradient search algorithm, named Delta rule (also Widrow-Hoff learning rule), was used for the automatic specification of interconnection strengths.

Interest in neural network activity dramatically increased with the discovery,[8] by Hinton, Rumelhart and McClelland (**Rumelhart**, **Hinton & Williams**, 1986; **McClelland**, **Rumelhart**, *et al*, 1986a) of a learning rule, based on an error back-propagation, for multi-layered perceptrons· this rule, has been shown to follow the same principles as Delta rule (**Matheus & Hohensee**, 1987). Hopfield's discovery of the applicability of neural networks to associative memories (**Hopfield**, 1982) triggered interest in neural networks among physicists (leading to a reduction of neural networks to mean-field theory, **Amari** (1983), Ising spin systems, **van Hemmen** (1986), **van Hemmen**, **Grensing**, *et al* (1988a, 1988b) and attractor dynamics, **Amit** (1989)) and the signal processing community.

By then, **Kohonen** (1977, 1980, 1984), **Fukushima** (1969, 1970, 1980), **Fukushima & Miyake** (1982), **Grossberg** (1980, 1982) and **Carpenter & Grossberg** (1986a) had made substantial contributions to the usage of neural networks in visual signal processing: however, these got to be widely recognized only after the publication of inves-

[8] The controversy related to the discovery of learning in multı-layer neural networks is traced by **Olazaran** (1993)

tigations by Rumelhart and Hopfield. As neural networks regained acceptance as a viable computational paradigm, focused inquiries into their capabilities have been pursued. **Valiant** (1984) has addressed the question of learning, though as a non-empirical enquiry. The issue of learning complexity has been investigated by **Judd** (1990). Learning with generalization has been the focus of investigations by **Valiant** (1984), **Baum & Haussler** (1989) and has been related to the Vapnik-Chervonenkis dimension (**Hertz, Krogh & Palmer**, 1991) in addition.

Connectionism, in its new form, attracted attention not only at the theoretical and procedural issues of processor representation, but also at the level of hardware realization. Mead, the pioneer of VLSI, proposed schemes for the realization of 'analog VLSI' (**Mead**, 1989; **Mead & Ismail**, 1989) and designed electronic retina and cochlea (**Lyon & Mead**, 1990), to emulate the human visual and auditory sensations. In view of the fact that connectionism seeks for knowledge representation in inter-processor interconnection strengths and the individual processors are very small (owing to simplicity) and VLSI technology too mandates similar requirements, VLSI implementation of processors with a neural basis leads to efficient wafer utilization. Neural networks, specially of the dynamical kind, have been employed in the routing of (conventional/symbolic) VLSI processors (**Jayadeva**, 1993).

In view of practical difficulties faced in the training of neural networks, typically the long sessions of training, enormity of processing nodes for realistic problems and the general inability to easily correlate rules and features to individual nodes, hybrid approaches, taking on a mix of concepts from the symbolic and connectionist traditions have been investigated in the literature. This approach, strongly inclined towards a Cartesian dualism, views neural networks as a pre-symbolic computational substrate providing the necessary interface between the 'external' world and the symbolic computational units.

1.2 An Overview of the Thesis

Neurons, in the present investigations of connectionist information processing, are operational generalizations of the threshold and decision units of Rosenblatt's perceptrons and representation of knowledge available (or given) through examples of association, continues to be sought through a storage of information in the connection strengths between processing units: the information stored is unrelated to the individual patterns incident on the 'retina'. However, the information stored in the connection strengths, also known as weights, is related to the functional dependencies specified – the specification is, in general, not exhaustive – through examples of valid association. Though this aspect has been appreciated and used, to advantage, in current investigations, no study seems to explicate the nature of preservation.

Recognition of patterns and function approximation (signal/state estimation) being the core of information processing, especially in automated systems, the limitations of processor realization in single neurons has necessitated processing schemes involving an interconnected ensemble of neurons. Processing schemes involving layers of neurons are typical, the interconnections being between neurons in different, generally adjacent, layers and, in selective cases, between neurons of the same layer. While significant claims have been made, in the literature, regarding the adequacy of layered neural networks, with a single or multiple layers of decision making, in the recognition of patterns and function approximation, a relative evaluation of the representational effort needed in these networks does not seem to have been considered.

Present research on neural networks, inspired by the requirements of automating intelligence, has resulted in a plethora of networks, each concentrating on capturing specific aspects of input-output relationships. Discussions of these networks are also accompanied by elaborate procedures for the automated specification of processor parameters (*ie*, learning, or automatic programming) However, in this collective enquiry, as much theoretical as empirical, the axioms governing the study, are not generally stated. While it would be incorrect to state that connectionism is merely a collection of *ab initio* axiomatic statements and the ensuing logical discourse, it would be inappropriate not to seek the logical basis underlying the structure of current thinking.

The notion and nature of representation in neural networks have not been explicitly stated, though, a common theme discernible in the literature is that neural networks accommodate a localized representation of the input patterns at the output. This facet of neural networks has been, time and again, referred to as the ability of providing a plausible account of concepts induced on feature space Connectionism, in the currently available literature, is generally regarded as being restricted to realize functions between numerical spaces, though there seems to be no obvious reason why such a restraint should be operating. Indeed, the spirit of connectionism seems to exclude realization of functionals, operators, relations and mappings between more general spaces: one of the far reaching implications of this restriction is to miss the opportunity to automate (in a neural basis) the decisions (information processing) related to the design and operation of neural networks.[9]

In this thesis, I consider a characterization of the representation of signal processors with neural networks, under the topic of connectionist signal processing systems: however, I do not lay claim towards exhaustiveness in the study reported herewith. Representation is interpreted, in this thesis, as a decomposition and/or synthesis of the desired function through 'basis' functions that are not chosen *a priori*, but are synthesized to suit the requirements of processor realization: the requirements are specified through finitely many 'examples' of the

[9] If connectionism is extended to spaces more general than numerical, statements regarding neural networks, essentially mappings between function, functional, or operator spaces, could possibly be captured in neural networks. This ability would pave the way for a definition of a universal neural network, through which a study of the limitations of connectionism in cognitive modeling and proper comparison (possibly a unification) between connectionist and classical AI could be thought of

desired association. The 'basis' functions are, however, not restricted to be dependent on the family of functions under consideration.

Preservation of information in the processor realization offered by individual neurons, representational features of layered networks of neurons, paradigmatic concerns of representing signal processors with neural networks and characteristics of representation, in signal processing terms, offered by interconnected ensembles of neuron layers are the four broad topics that have been investigated. The principal claims of the study are listed below.

(a) Every weight associated with a neuron, in the sense of channeling information from different sites in a network (including elements of the incident input pattern) to the decision component of the neuron, preserves mappings on certain discrete pattern collections as sequences, this preservation reduces learning to an enumeration of weights – the enumeration is not without structure – and a search for threshold in a linearly ordered space.

(b) Multi layered neural information processors, independent of the degree of layering, are adequate for representing functions on preservance input spaces, the demand on the number of processing nodes decreasing, in general, with the number of nodes: this adequacy translates into an assurance of the possibility of extending connectionist information processing to symbol spaces wherein linear separability, a basic characteristic required of the processing nodes, is stated algebraically as the partitioning induced by a dichotomy on a (semi) lattice such that each member of the partition is a semi-lattice.

(c) Processor representation is achieved in neural networks through point-wise nonlinear associations between integral transforms; the kernels, generally nonlinear, of the integral transforms when synthesized through neural networks allow an incorporation of *a priori* knowledge of the processing architecture and functionality

(d) Function representation in neural signal processors is accompanied by localization and concepts, identified as processor responses, reflect a restriction of evaluation, expressed as a weighted average of representations in wavelet frames, to localized regions in the sheaf of input patterns.

On learning the motivations for the investigations reported herein, it is now imperative to know the salient queries that need to be addressed in order that the study be satisfactory. Representation of processors in neurons being of a primary nature, answers to the following queries would enable an understanding of function representation in ensembles of interconnected neurons.

i. Does there exist assignments to the connection strengths in a neural network that ensure a preservation, of relative order between inputs, in the representation of functions?

ii. If connection strengths of the kind indicated above do exist, what is the extent of the input pattern space that is subject to the criterion of preservation?

iii. How does the existence of connection strengths that preserve relative order between inputs affect the issues in function representation, especially learning and generalization?

Preservation of relative order between inputs has been considered as the basis of the above enquiry noting that partitioning effected, on the input space, by decision elements, based on an approach of comparison (of discriminants with a threshold), is decided to a large extent by the relative ordering enforced on the inputs in the process of evaluating the discriminants. Representational schemes that can be shown to be generalizations of positional numbering have been shown to provide preservation in the sense mentioned above and I have established the existence, with an identification, of certain discrete spaces, corresponding to each non-null configuration of connection strengths, that accommodate a preservation in function representation. A reduction of learning to the distinct steps of weight (connection strengths) enumeration and a search for threshold in a linearly ordered space has been shown. The interplay of the issues of generalization and learning in the selection of weights and threshold are also indicated, though cursorily.

Assurance of the existence of assignments to connection strengths that allow a preservation, of relative order between inputs, in function representation leads to an interest in the manner in which the feature of preservation can be used to understand the nature of function representation in layered neural information processing systems. In this connection, the following enquiry would be helpful.

i. How does preservation, of relative order between inputs, affect function representation in layered neural information processors?

ii. How is the representation of functions in layered networks of neurons affected by the degree of layering?

iii. Given that preservation, of relative order between inputs, is applicable on certain discrete spaces does there exist an interpretation of the processing functionality that would allow an extension of neural computation to symbol spaces?[10]

I have established that while the scope of function representation, discussed in the restricted context of functions on the discrete spaces

[10] Due to the much acclaimed success of neural networks and the projected differences between symbolic and sub-symbolic (neural) approaches to information processing, it is important to know whether, or not, the paradigm of neural networks is restricted solely to approximation of functions on continuous (numerical) spaces, *ie, is it ever possible to extend [naturally] neural processing to functions defined on abstract spaces, typically symbol spaces?* In order that the insight we try to get of the representational paradigm in artificial neural networks is non-trivial, an essential requirement is that the notion of representation should be applicable independent of the nature of neurons used in processor realization. This notion, in addition to providing a basis for acquiring a unified understanding of neural signal processing schemes, could be useful in exploring the possibility of using the neural paradigm in the decision making and search problems related to neural networks· this, if successful, would project the neural computational paradigm as an alternative to *formal automata*, a typical example being *Turing Machines*.

in which preservation (of relative order between inputs) holds, is really unaffected by the degree of layering, the nature of representation varies in the sense that an increased degree of layering leads, in general, to a realization of the given function with fewer processing nodes than necessary in the case of processors involving a single layer of neural decision elements. An alternative definition to linear separability, the basic processor characteristic in most neural networks, has been established: I reproduce the definition.[11]

> *A dichotomy on a (semi) lattice is said to be linearly separable if the [embedding] lattice can be expressed as a partition, each component of the partition being a semi-lattice.*

Concerning the representation of signal processors with neural networks we are faced with the following essential questions.

i. What plausible axioms are necessary for a discourse in neural signal processing?

ii. What operational interpretation would allow for a unification of existing neural architectures and suggest novel architectures?

iii. What is the nature (and characteristic) of representation in neural signal processing?

[11] This definition allows us to appreciate that the spirit of connectionism need not be restricted to mappings between numerical spaces.

These questions refer to the nature of information storage and handling in neural networks. I have shown that four axioms are essential to neural signal processors: these are reproduced below. (The axioms described below are related only to the operational character of neural signal processing and do not stipulate either the components or the context in which such processing is realized.)

1. **Axiom of Organization.**

   A neural signal processor is composed of (layers of) three operational stages: measurement, discrimination and aggregation in that order. Preprocessing, if any, (preceding, or incorporated in, the measurement) is sought to be represented in a neural basis. Measurements are effected on an observation space constructed as the Cartesian product of the input space and a relevant subspace of a union of the space of responses of the distinct layers.

2. **Axiom of Measurement.**

   A neural signal processor, through the measurement functions in each of the processing (decision making) nodes, induces a foliation, of codimension at least one, in the input manifold. This foliation forms the basis of synthesizing (approximating) the desired level curves of the function.

3. **Axiom of Discrimination.**

   A neural signal processor, through its discriminatory functions, renews the foliations, induced on the input space by the measurement functions, through a transformation, of the stems of the foliations, with at least one of the following properties:

(a) alter the indexing of leaves to retain distinctness in a finite non-zero number of local regions of the input space,

(b) introduce multiple components in the leaves,

(c) associate, to at least one component of a leaf of the foliation due to discrimination, uncountably many leaves of the foliation due to measurement.

Re-foliations provide the basis for establishing equivalences between members (elements) of the input space in ways not possible through the chosen measurement functions.

4. **Axiom of Aggregation.**
A neural signal processor, through its aggregation function, synthesizes (or approximates) the level regions of processor response through a foliation on the Cartesian product of the stems of foliations on the input space due to discrimination. Concepts, in neural signal processors, are identified with the level regions of processor response.

From the signal processing perspective, neural signal processors, viewed as integral transforms interacting via point-wise nonlinear transformations provide the key to unify the several architectural types.[12]

[12] The motivation for seeking a unified understanding can be seen in the following

Indeed, the interpretation of neural signal processing as nonlinear transformations between integral transforms relates nicely to the view that decisions are taken on feature spaces and that feature extraction (also termed pre-processing) can be sought to be realized in a neural basis, in a manner similar to the feature processing. Neural signal processors with sigmoidal activation functions effecting nonlinear as-

perceptive remarks. (All these statements are found in **Machlup & Mansfield** (1983a), p. 7–8)

> Several analogies have been used to characterize isolationist or parochial attitudes of specialists uninterested in cognate or complementary fields of inquiry. For example, they erect fences around their fields – like unsociable property owners inhospitable to their neighbors.
>
> [F]ields of scientific work ... which have been explored from the different sides of pure mathematics, statistics, electrical engineering, and neurophysiology, in which every single notion receives a separate name from each group and in which important work has been triplicated or quadruplicated, while still other important work is delayed by the unavailability in one field of results that may have already become classical in the next field. [A case in point is the discovery of algorithms for learning in multi-layer perceptrons]
>
> It is these boundary regions of science which offer the richest opportunities to the qualified investigator. (**Wiener**, 1948, p. 2)
>
> [S]cience is split into innumerable disciplines continually generating new subdisciplines. In consequence, the physicist, the biologist, the psychologist and the social scientist are, so to speak, encapsulated in their private universes, and it is difficult to get one word from one cocoon to the other. (**von Bertalanffy**, 1968, p. 30.)
>
> The Republic of Learning is breaking up into isolated subcultures with only tenuous lines of communication between them ... an assemblage of walled-in hermits, each mumbling to himself words in a private language that only he can understand. (**Boulding**, 1956, p. 198.)
>
> However, in his plea for interdisciplinary collaboration, Boulding warned that "it is all too easy for the interdisciplinary to degenerate into the undisciplined." (*Ibid*, p. 13.)

sociations on linear discriminants have been shown to be adequate to represent all the architectural novelties suggested by the axioms of neural signal processing. This investigation, however, provides an insight into the nature of representation in a superpositions of functions, each related to the other through a permutation of weights. Such superpositions, in neural signal processors with sigmoidal activation functions, have been shown to realize functions necessitating activation functions that are non-sigmoidal. I have also indicated that the kernels, generally nonlinear, used in neural signal processors when realized through neural signal processors involving multiple layers relate to issues involving the incorporation of *a priori*, but partial, knowledge about the interconnection strengths between processors.

The local nature of representation characteristic of neural signal processors, discussed earlier, motivates the following enquiry. Note that this investigation proceeds in the framework of neural signal processors being point-wise transformations between integral transforms.

i. How do the kernels of integral transforms and the mechanism of nonlinear association influence the nature of localization in function representation?

ii. What, in terms of processing of signals, are the characteristics of localization in the neural approach to function realization?

iii. What is the implication of localization on the nature of information processing realized through neural signal processors?

Localization in neural networks induced, by considerations of realization in the connection strengths (and thresholds), due to the kernels of integral transforms have been shown to be related to the predicates evaluating relative organization of assignments within a pattern: this aspect of localization has been qualified through the term *intra-pattern* predicates. In contrast, localization due to the mechanism of nonlinear association relates to predicates evaluating relative organization of assignments between patterns, this aspect has been qualified through the term *inter-pattern* predicates. In addition, concepts represented by nodes of neural (signal) processors have been shown to be localized regions in the sheaf of patterns, each concept being the consequence of a conjoint evaluation of intra-pattern and inter-pattern predicates. This latter statement suggests that representation in neural signal processing is neither localized in the sense of individual nodes being identified with distinct concepts nor has an involvement of the entirety of nodes in a network participating in the synthesis of a concept.

Kernels of the reproducing type, as a choice for the integral transforms of measurement and aggregation, has been considered in the localization of representation in neural signal processors. Such a localization shows that the nature of representation is in the sense of the measurements effecting a reconstruction of the incident (local) concept through finitely many (non-uniformly spaced) samples. The synthesis of 'basis' functions, in neural signal processors, that support a representation of the desired processor has been shown to be in the sense of the measurements effecting a representation of the incident concepts in the basis functions that are used to realize the kernels of the integral transforms of measurement and the responses (aggregates) effecting a representation of the decisions (discriminations) on the measurements in the basis functions that realize the kernels of the integral transforms of aggregation. Such a representation has also been related to the notion of representation in wavelet frames.

1.3 Organization of the Thesis

Chapter 2 is devoted to a review of signal processing with neural networks, the central theme of this thesis. The review is divided into four components, each corresponding to a separate section. Signal processing, at a reasonably abstract level of formulation and some of the established approaches to signal processing are considered in § 2.1 (*p*. 35). This section also attempts to bring out the importance of signal and system representation and also trace some of the key requirements of signal processors. A review of artificial neural networks is taken up in § 2.2 (*p*. 54). An abstract formal model of single neurons and specific (existing) models of interest captured by the abstraction present the background in which a study of the need for networks of neurons and the architectures of neural networks are studied.

The abstractions of signal processing and neural networks, are utilized in § 2.3 (*p*. 82) dwelling on a review of neural signal processing.

An effort is made to provide a glimpse of the history of neural signal processing. This is not irrelevant especially in a context wherein the attempts to process signals with neural networks have been initiated soon after the invention of perceptrons but have attracted attention only since the mid-1980's – a period which also saw furious debates on the nature and relevance, of artificial intelligence, with serious attacks on classical and to a lesser extent, connectionist approaches to AI.

A study of processor representation in isolated neurons has been presented in Chapter 3. The existence of assignments to connection strengths that allow a preservation, of relative order between inputs and a characterization of the input subspaces that accommodate this preservation has been established in § 3.1 (*p*. 110). Function representation with preservation and an appraisal of linearly separable functions, the only dichotomies represented by neurons with binary comparators, has been considered in § 3.2 (*p*. 138). Learning, in a context where preservation is supported, is reinterpreted in § 3.3 (*p*. 154) and the interplay between learning and generalization is indicated. In § 3.4 (*p*. 171) the notion of preservation is extended to numbering systems with radices different from binary and an identification of preservance input spaces – input spaces preserved by any arbitrary, but non-null, assignment to connection strengths – has been sought.

I introduce, in Chapter 4, the notion of neural signal processing and investigate the influence of preservation, of relative order, on function realization. § 4.1 (*p* 188) is a discussion of function representation on single layered neural signal processors. Multi-layered varieties are investigated in § 4.2 (*p*. 201) from the perspective of function representation in the context of preservation. In § 4.3 (*p*. 208) I discuss the issue of identifying preservance input spaces appropriate to a given training set, an equivalent of learning the weights of the first layer. § 4.4 (*p*. 218) introduces an algebraic equivalent of the notion of linear separability and concludes with a cursory look into the possibility of representing functions between symbol spaces.

Representational issues in neural signal processing architectures form the theme of Chapter 5. Neural signal processors, with types, are defined in § 5.1 (*p*. 237). The potential for representation in neural signal processors is also investigated in this section. I establish, in § 5.2 (*p*. 249), the functional characteristics of neural signal processors and state the axioms of organization, measurement, discrimination and aggregation. Neural signal processing has been interpreted in § 5.3 (*p*. 276) as involving point-wise nonlinear associations between integral transforms: these transforms relate to measurement and aggregation operations. In § 5.4 (*p*. 284) representation in neural signal processors has been considered from the perspective of function approximation.[13]

Localization in the functions represented by neural signal processors has been investigated in Chapter 6. In § 6.1 (*p*. 316) I have discussed the

[13] Note that the essential nature of intelligence consequent on information processing reduces to approximation of functions describing the relevant class/region memberships.

influences of kernels on localization, this discussion has been developed in the context of isolated neurons. An extension of the kernel influence on localization in functions represented in layered neural signal processors accompanied by an investigation of the localization influenced by the mechanism of nonlinear association forms the discussion of § 6.2 (*p*. 322). A characterization of localization in terms of signal processing and the implications of localization on the nature of processing effected by the neural approach to information processing have been considered in § 6.3 (*p*. 332). The influence of kernel structure on the representation in neural signal processors has been studied in § 6.4 (*p*. 344).

In Chapter 7, the final chapter, I sum up the conclusions of the investigations in the preceding chapters The relevance of some of the key results and interesting directions of further study have also been incorporated in this chapter. In addition, two appendices have been included. Appendix A provides a glimpse of the prominent approaches suggested for an automation of intelligence. The notations used in this thesis have been listed in Appendix B. A list of references cited in the thesis has been included after the appendices.

Chapter 2

Signal Processing with Neural Networks: A review

So what comes of our making,
Is slices of history pieced together,
In full knowledge that
All history is distortıon of reality,
And appropriate fillers added to reinforce
The image of what we want it all to be in retrospect,
At best a Romance of sorts.

And it's a Romance that keeps us going,
Sometimes asunder.

— Weepy Sinner (Prof. V P Sinha)
ın *Hıstory and Romance: A Joem or a Poke*,
Indian Institute of Technology Kanpur

Recently neural networks, a class of hierarchical nonlinear dynamical systems, are the focus of attention in connection with the realization of nonlinear signal processors. In several instances of signal processing operations, *eg*, recognition of handwritten characters, facial recognition, texture identification, sonar signal classification, speech signal processing, *etc*, parameterization or grammatical formulation of the signal space, approaches common to the conventional approaches of signal processing, is not amenable due to insufficient understanding of the processes underlying the signal generation and/or the volume of (empirically observed) data being grossly insufficient to allow (reliable) closed-form input-output relationships from being established. At a lay level of interaction, such operations are believed to involve a subjective element in the processor.

These operations, however, have finitely many examples of the input signal, *ie*, prototypes, for which the corresponding outputs are known either completely or partially. It is of interest to realize the corresponding processor in such a manner as to extract invariances, related to the processor, from the finite number of examples and incorporate the extracted invariances while estimating the processor output corresponding to input signals not included in the repertoire of prototypes. Neural networks, essentially computational models inspired from current accounts of (human) cognitive abilities and grounded in the accumulated understanding of the biological substrate supporting information processing, present a framework supporting both requirements: the process of extracting invariances is addressed, as a representational issue, through problems of learning and incorporation of (extracted) invariances in processing is formulated as a problem of generalization.

In this chapter, I review, cursorily, the status of current research related to signal processing with neural networks, to indicate the relevant background, terminology and notations: the approach has been to focus more on the issues important in neural signal processing, than on enumerating specific accomplishments of processing with neural networks. Abstractions in understanding signals, their processing and classification of processors are focused in § 2.1. In § 2.2 (*p*. 54), artificial neural networks are reviewed preparatory to an understanding of neural signal processing. The history of neural network based signal processing and current understanding in architectures, algorithms and usage of neural networks are considered in § 2.3 (*p*. 82).

2.1 Signal Processing: Crucial issues and necessities

Processing of information is an essential requirement of automation, in particular, the mechanized expression of intelligence and signals provide a vehicle (or medium) for expressing the desired representations of entities, events and objects in the physical world (also termed reality). Such a representation is necessitated by the requirements of information manipulation, symbolic or otherwise. As signals and the information they convey acquire an ontological status and get to be recognized as valid members of the physical world, thereby necessitating information, in addition to matter and energy, as an essential dimension of material manifestation, signals could, indeed, be representing properties, traits, or qualities of materials, including signals.[1]

Signals are commonly expressed as functions (processes) describing the entity or object under consideration as a dependence on a narrow, localized, region of the space-time continuum we are accustomed to term as the *Universe*: it is common, though not essential, to use numerical assignments, or assignments involving vectors of numbers, to the domain and range of the signals. The functions, rather than being arbitrary, are expected to conform to the physical and/or biological constraints, if any, involved in the process of sensation

In this view, we have visual sensations described as a matrix of numbers, auditory sensations described as a sequence of numbers, *etc*, as valid examples for signals. Commonly real-numbers and to a certain extent complex numbers, are used in encoding the domain and range values of signals. With advances in digital processing technology, recent efforts aimed at symbolic encodings for domain as well as range values and associated algebraic structure of processors attempt to exploit the symbolic processing methodology.

[1] While it is expected of signals to be used as a meta-language describing objects in the physical world, it is not incorrect to include signals in the object language, particularly in the context of the abstract notion of material introduced in the previous chapter.

The information content of a signal is sought in the relative organization of assignments over the domain and in this sense, signals are the means of information interchange between communicating processes. In classical AI, the relative organization is expressed as a predicate of an appropriate mode of logic, while in statistics and connectionist AI, descriptions of relative organization are to be found in the signal statistics (distributions): incidentally, the two are not unrelated and in this thesis I will use the term *predicate* to mean such a relative organization. One of the key requirements in the processing of information (pattern recognition) is to identify the predicates relevant to the task at hand,[2] and to detect the predicate(s) applicable to the incident signal

In the processing of signals, it is common to find that signals received from processes separated in space and/or time are not identical to the original or intended one, thereby the relative organization of assignments in the received signal could be different from that in the original signal. Under these circumstances, signal processors, in order to satisfy the information processing requirement, are equipped with a distance measure, with which predicates corresponding to the incident signal are evaluated with the repertoire of relevant predicates (possibly updated in the progression of time) and the signal is classified on the

[2] In classical AI the identification of relevant predicates (hypotheses) is formulated as the problem of (knowledge) representation, while in neural networks, the same is accomplished in the learning phase/mode.

basis of this evaluation, generally in the sense of least deviation, which is based on the topological notion of continuity.[3]

Signals and their Processing

A signal processor, purporting to *generate*, or *extract*, a signal y from a (given) signal x, essentially scans x over a subset of its domain and in each scan, based, in general, on a fixed and/or finite number of arguments, derived, from the signals x and y, in accordance with the scan position, evaluates the assignment to the signal y, at the corresponding scan position, as a function of the arguments (derived

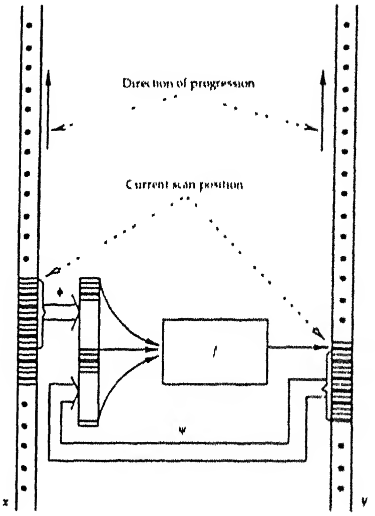


Figure 2.1: Essential structure of a signal processor

[3] Signal processing is also desired in the sense of mapping the incident signal to one which has its assignments relatively *better* organized. This problem, essentially similar to that of signal classification, addresses the issue of extracting information (in the sense of predicates) from a given signal and processors employed for this purpose are termed *filters*. Estimation of signals too is based on similar principles and hence, in the ensuing discussion, I will use the terms filters, processors and estimators interchangeably so as not to lose generality.

from x and y). Symbolically, the preceding statement can be expressed concisely as in the following. An illustration too accompanies in Figure 2.1 for clarifying the statement.

$$\begin{array}{rcr}
 & x.\ \Xi \to \mathcal{X}, \quad y.\ \Theta \to \mathcal{Y}, & \text{(2.1a)} \\
 & \forall\theta \in \Theta \quad \exists \mathcal{N}_x(\theta) \subseteq \Xi,\ \mathcal{N}_y(\theta) \subseteq \Theta & \\
\text{such that} & \mathcal{N}_x(\theta) \cup \mathcal{N}_y(\theta) \neq \emptyset, & \\
 & \mathfrak{a}_x(\theta) \triangleq x|_{\mathcal{N}_x(\theta)} \in \mathfrak{A}_x,\ \mathfrak{a}_y(\theta) \triangleq y|_{\mathcal{N}_y(\theta)} \in \mathfrak{A}_y, & \\
\text{and} & y(\theta) = f(\phi(\mathfrak{a}_x(\theta),\theta),\psi(\mathfrak{a}_y(\theta),\theta),\theta), & \text{(2.1b)} \\
\text{where,} & \phi\colon \mathfrak{A}_x \times \Theta \to \mathfrak{B}_x, \quad \psi\colon \mathfrak{A}_y \times \Theta \to \mathfrak{B}_y, & \\
\text{and} & f \cdot \mathfrak{B}_x \times \mathfrak{B}_y \times \Theta \to \mathcal{Y}. &
\end{array}$$

In the above expressions, I use the following notations.

Ξ := Domain of definition of the signal x Scanning of the signal over this domain is indicated by the progression of $\xi \in \Xi$.

Θ := Domain of definition of the signal y Scanning of the signal over this domain is indicated by the progression of $\theta \in \Theta$.

$\mathcal{X}$:= Range space of the signal x.

$\mathcal{Y}$:= Range space of the signal y.

$\mathfrak{A}_x$:= Algebraic structure (of appropriate kind) on $\mathcal{X}^{\Xi}$, the space of all signals from Ξ to $\mathcal{X}$. ($\mathfrak{A}_x$ contains subsets of $\mathcal{X}^{\Xi}$)

$\mathfrak{A}_y$:= Algebraic structure (of appropriate kind) on $\mathcal{Y}^{\Theta}$, the space of all signals from Θ to $\mathcal{Y}$. ($\mathfrak{A}_y$ contains subsets of $\mathcal{Y}^{\Theta}$.)

$\mathfrak{B}_x$:= Space of measurements on signal x. (Possibly the same as $\mathfrak{A}_x$.)

$\mathfrak{B}_y$:= Space of measurements on signal y. (Possibly the same as $\mathfrak{A}_y$.)

$\mathcal{N}_x(\theta)$:= Neighbourhood structure in Ξ at the scan position θ, $\mathcal{N}_x(\theta) \subseteq \Xi$ for all $\theta \in \Theta$.

$\mathcal{N}_y(\theta)$:= Neighbourhood structure in Θ at the scan position θ, $\mathcal{N}_y(\theta) \subseteq \Theta$ for all $\theta \in \Theta$.

$\mathfrak{a}_x(\theta)$:= Assignments of x over $\mathcal{N}_x(\theta)$. (Note that $\mathfrak{a}_x(\theta)$ is a signal in $\mathcal{X}^{\mathcal{N}_x(\theta)}$ and also identifies an ordered subset (ordered by $\mathcal{N}_x(\theta)$) of $\mathcal{X}^{\Xi}$ such that $\forall\theta \in \Theta\ \mathfrak{a}_x(\theta) \in \mathfrak{A}_x$.)

$\mathfrak{a}_y(\theta)$:= Assignments of y over $\mathcal{N}_y(\theta)$. (Note that $\mathfrak{a}_y(\theta)$ is a signal in $\mathcal{Y}^{\mathcal{N}_x(\theta)}$ and also identifies an ordered subset (ordered by $\mathcal{N}_y(\theta)$) of $\mathcal{Y}^{\Theta}$ such that $\forall\theta \in \Theta\ \mathfrak{a}_y(\theta) \in \mathfrak{A}_y$)

ϕ := Indexed collection of measures[4] on $\mathfrak{A}_x$, indexed by $\theta \in \Theta$.

ψ := Indexed collection of measures on $\mathfrak{A}_y$, indexed by $\theta \in \Theta$.

f := Mechanism (method) by which the evaluation of assignments to y are arrived at This includes *correlations* between the signals x and y.

The above formal statement[5] is sufficiently general to allow each of Ξ, Θ, $\mathcal{X}$ and $\mathcal{Y}$, to be either discrete or continuous, numerical or symbolic and scalar or vector collections (generally vector spaces) and captures the essential traits of nearly all of the distinct signal processor

[4] More precisely, ϕ and ψ are product measures on the algebraic structures $\mathfrak{A}_x$ and $\mathfrak{A}_y$ respectively These functions measure an appropriate (desired) aspect of the relative organization of assignments in the signals x and y, *ie*, ϕ and ψ are predicates compatible to signals x and y.

[5] In the present form, this statement dictates the processing model to be of the *data driven* kind, *ie*, processing is initiated only on the relevant portion, *viz* $\mathfrak{a}_x(\theta)$ and $\mathfrak{a}_y(\theta)$, of the signals x and y being available. (See **Gorsline**, 1986, for a discussion on the data driven computational model.) The functional relationships easily suggest correspondences, if not an isomorphism, with the formalism of Turing Machines, common in the theory of computation.

kinds Signals are abstracted as functions (or processes) and the role of a signal processor is to establish associations between such abstract entities as depicted in the following.

$$\begin{tikzcd}
\mathcal{X} & \mathcal{Y} \\
\uparrow x \arrow[r, mapsto, "f"'] & y \uparrow \\
\Xi & \Theta
\end{tikzcd} \tag{2.2}$$

Signal processors, as described above, belong to the class of *abstract dynamical systems* when $\mathfrak{A}_y \neq \emptyset$, and the influence on y of the assignments $\mathfrak{a}_y(\theta)$, for all $\theta \in \Theta$, through the map ψ is not null.

Contextual information contained in signals, x and y is formally captured through the algebraic structures $\mathfrak{A}_x$ and $\mathfrak{A}_y$ respectively, and the functions ϕ and ψ indicate the nature of evaluation (measurement) of information content in signal regions specified by the neighbourhoods $\mathcal{N}_x(\theta)$ and $\mathcal{N}_y(\theta)$ respectively. It is not uncommon to find the algebraic structures $\mathfrak{A}_x$, $\mathfrak{A}_y$ and the space of measurements $\mathfrak{B}_x$, $\mathfrak{B}_y$ being of the nature of (algebraic) fields. Note that this formalism encourages a recursive (at times, circular) understanding to signal processors, *ie*, each of the components of the above formalism, specifically ϕ, ψ and f, are valid candidates for being considered as signal (information) processors. It is also important to note that specification (and the structure) of neighbourhoods $\mathcal{N}_x(\theta)$, $\mathcal{N}_y(\theta)$ and functions ϕ, ψ and f, are crucial to signal processor design, specification and classification. These issues translate to those of signal and processor representation noting the recursiveness involved in the understanding of signal processors.

Requirements of Signal Processors

Signals are functions aimed at a representation of perceptual entities in the 'reality' around us. The role of signal processors is to enhance, or protect from possible deterioration, the perceptual qualification of signals obtained from data gathered through sensors, or measurement apparatus, specially in situations involving signal translocation through space and/or time,[6] possibly through media introducing distortions.

In this role, processing is viewed as a means for guiding perceptual categorization and, in the context of automation (including that of intelligence), is expected to reduce the cognitive burden of human participants. This latter requirement, typical in situations involving the control/coordination of complex systems, immediately translates to an endeavor seeking for a normalized performance of the processors, the criterion of normalization being the *satisficability*[7] of the outcome of processing to 'average' participants in the efforts of automation.

[6] Translocation of signals through space (and, inevitably, in time) forms the crux of communication (see **Cherry**, 1957, for the essential nature of the problem of communication, and a glimpse of its theoretical structure) and is commonly studied as signal transmission. Signal transportation in time, without appreciable variation in spatial location, is commonly experienced in situations involving storage and retrieval. In both of the extreme forms, physical considerations disallow distortion free signal handling.

[7] Satisficability refers to the exigency of compelling the processor to provide the desired and/or idealized performance within tolerable limits. While this requirement could be mathematically formulated as a minimization, within the limits of tolerance, of the deviation of processor performance from that desired, or idealized, an inevitable element of subjectivity is encountered in deciding the limit of tolerance. The term satisficability has been chosen to signify, what has come to be known as, the subjective element involved in approximation. I acknowledge Prof VP Sinha for introducing me to this term.

A requirement additionally encountered in the design of processors is to relate the processing steps to prevailing conceptual categories, typically feature extraction (incorporating dimension reduction), feature discrimination and concept aggregation, of information processing by (average) humans: such a signal processing mechanism is then touted as a model for mental activity (intelligence). In passing, it is important to note that the average (or stereotypic) human participant in information processing is only a postulate and no candidate need be expected to satisfy the criterion of averageness.

The criterion of satisficability to 'average' individuals is commonly expressed mathematically in terms of function approximation. Perceptual entities, as mentioned earlier, are conveyed (represented) in signals through the relative organization of assignments over relevant domains. In terms of the formal statement presented in the preceding article, the localized assignments $a(\cdot)$ relate, in terms of structural constraints (as measured by ϕ and ψ on signals x and y, respectively), to perceptual entities.

Function realization being the essential nature of signal processors, the context in which the realization is being sought influences the requirements of processing. While function realization translates to a requirement of exact reconstruction in signal processors between symbol spaces, the requirement in signal processors defined between continuous spaces of numbers is always one of obtaining an approximation:

both variants are relevant criteria of satisficabilty. Such a difference in processing requirements stems from the fact that while in continuous spaces of numbers a natural notion of proximity, or neighborhood, is appreciated regardless of the specific details of the space, no such universal notion can be associated with symbol spaces.

The (un)satisficability criterion governing the choice of assignments to the output signal given an input signal (*ie*, the design of the processing rule) is mathematically expressed in terms of a measure of mismatch – generalized distance – between the output signal and an idealized (or expected or desired) form of processing on the input signal: the measure of mismatch is designed, or chosen, to incorporate the specific aspects of the perceptual entities that need to be preserved in the signal processing operation. Referring to the notations in the formal statement in the previous article, the symbolic form of the (un)satisificability criterion is

$$\min_{y \in Y^{\Theta}} \|\mathfrak{s}(y(\theta)\,,\mathfrak{a}_y(\theta)\,,\theta)\|_{\theta\in\Theta}\,, \tag{2.3}$$

$$\text{where,} \qquad \mathfrak{s}\cdot \mathcal{Y} \times \mathfrak{A}_y \times \Theta \rightarrow \mathfrak{S}.$$

In the above expression, $\mathfrak{s}$ refers to the measure of mismatch, *ie*, (un)satisficability, $\mathfrak{S}$ denotes the repertoire of distinct labels (possibly numbers) used to distinguish the possible mismatches and $\|\mathfrak{s}(\cdot,\cdot,\cdot)\|$ denotes the accumulation of (un)satisficability over all the individual assignments (to signal y). The latter operation serves to express, through a single number (generally integer or real), a measure of (un)satisfiability for the entirety of the outcome of the processing operation: the desire to have this measure expressed as a single number stems from the need to order, given the natural ordering available in one-dimensional spaces, the possible outcomes of a processor for a given input signal and to select the best alternative, in the sense of minimum mismatch. It is not uncommon to find the operator $\|\cdot\|$ satisfying the axioms of a *norm* and this explains the associated notation.

The generic forms of the (un)satisfiability criterion designed to measure the mismatch of the output signal with an *a priori* specified desired signal, or an idealized (or expected) form of processing on the given input signal are, respectively, indicated in the following.

$$\mathfrak{s}(y(\theta), \theta) \triangleq \mathfrak{s}(y(\theta), \mathfrak{a}_y(\theta), \theta) = \rho(y(\theta), y_d(\theta)), \tag{2.4}$$

$$\mathfrak{s}(y(\theta), \mathfrak{a}_y(\theta), \theta) = \rho(y(\theta), g(\mathfrak{a}_y(\theta), \theta)), \tag{2 5}$$

where, y_d denotes the desired form of signal on processing, g is an appropriate function (possibly incorporating ψ) specifying the idealized (or expected) form of processing needed and ρ indicates the mechanism by which comparison between the output signal and the desired or idealized signal forms is achieved. In functional analytic terms, ρ is, generally, a (semi)metric, *ie*, a metric (distance function) with the axiom of unsignedness relaxed. Processors with the first form of (un)satisfiability criterion (*ie*, matching with desired signals) are termed *supervised* and those with the other form are termed *semi-supervised*.[8]

[8] Processors without explicit supervision, are, in general, termed *unsupervised*, though this term is not altogether appropriate

Automatic minimization of the measure of mismatch ($\mathfrak{s}$), formulated, equivalently, as a problem of search in the space of admissible solutions, is commonly addressed, specially in signal processing, as a variant of (stochastic) gradient descent and, consequently, $\mathfrak{s}$ is expected to be defined between continuous spaces exhibiting differentiability (almost) everywhere. The (un)satisfiability criterion is commonly formulated to have a *quadratic* variation in the numerical values of the desired and realized signal assignments (essentially minimization is on the L^2 – or ℓ^2 – norm[9] of error) and, from physicalist, considerations this criterion is given an interpretation of energy.

Though reasonably simple in concept, gradient descent based approaches have come in for sharp criticism due to an inherent (and inevitable) lack of speedy convergence and the undesirable feature of search seeking out 'locally optimal solutions' with a likelihood no less than that of seeking 'globally optimal solutions': locally optimal solutions are understood in the sense of the measure of the region around candidate solutions (*ie*, those meeting the satisficability criterion) relative to that of the space of admissible solutions. Convergence, equivalent to termination, of the search procedure is, in general, in a weak sense and can be assured only when the mismatch is evaluated as a quadratic function of the desired and realized signals.

[9] Note that $\mathfrak{s}$ is a characterization of approximation error and, hence, the error is expected to belong to a continuous space. However, the signal could be defined on a space which is either discrete or continuous: in the former case the term sequence is more frequently used. Minimization of $\mathfrak{s}$ is thus with respect to the L^2 norm when the signal definition domain is continuous, and ℓ^2 norm when discrete.

Several schemes, accompanied by claims of superiority over gradient descent, have been suggested to conduct (automated) search, the latest being *genetic algorithms* (**Goldberg**, 1989), distinguished, from other evolutionary approaches, in its applicability to abstract search problems, obviating the requirement of defining s over continuous spaces, or even requiring differentiability of the same. Relatively more convincing assurance of seeking globally optimal solutions and speedier convergence, despite the absence of a firm theoretical basis for such claims, characterize the approach of genetic algorithms Search being an essential component of training in (artificial) neural networks, traditional formulations of learning as variants of gradient descent in the space of (admissible) weights are currently being reinvestigated in the framework of evolutionary programming, typically genetic algorithms.

Signal Processor Types

Processors wherein the operations ϕ, ψ and f are all linear and the neighbourhoods $\mathcal{N}_x(\theta)$ and $\mathcal{N}_y(\theta)$, for all $\theta \in \Theta$, are imposed through delay, or shift, operations are termed *linear*: violation of any of these stipulations implies that the processor is *nonlinear*. If the measurement functions ϕ or ψ evaluate the signal assignments $a_x(\theta)$ and $a_y(\theta)$, respectively, depending on the position θ then a processor realized with such measurement functions is termed *adaptive*. Processors wherein the signal evaluation mechanism f is independent of θ are termed *shift-*

invariant, or *translation-invariant* and it is pointless to seek for such an invariance when adaptivity is incorporated.

If no element of random variation is incorporated in the construction of neighbourhoods ($\mathcal{N}_x(\theta)$, $\mathcal{N}_y(\theta)$) and functions (ϕ, ψ, f) then processors incorporating such components are termed *deterministic*, else *stochastic*. A new element of description is introduced in stochastic processors, that of *distributions*, *ie*, a characterization of invariances in the likelihood of relative organization of assignments in a signal, possibly in relation with other desired, or *a priori* chosen, signals.

The formalism in Equation 2 1 (*p*. 39) is capable of representing stochastic processors when the algebraic structures $\mathfrak{A}_x$ and $\mathfrak{A}_y$ are Borel (sigma) fields and when the functions ϕ, ψ and f, which are to be designed to capture the relevant (desired) distributions, satisfy the axioms of probability measures. In such processors, the members of $\mathfrak{A}_x$ and $\mathfrak{A}_y$ are commonly termed *events*. Shift invariance is generally considered in terms of *stationarity*, however, this invariance is qualified by the particular distributions of interest in view of the fact that processing is characterized by distributions.

Signal processors for which $\Xi \equiv \Theta$ are termed *filters*: in this class the input and output signals (x and y respectively) are described on the *same* domain. Processors functioning as filters are termed *causal*[10]

[10] Causality is, incidentally, natural only when the signal definition domain is one-dimensional and has the interpretation of time.

if the common signal domain Θ ($\equiv \Xi$) is a partially ordered set (with respect to an appropriate (precedence) relation denoted by $\preceq$) and for all $\theta_1, \theta_2 \in \Theta$, $\theta_1 \preceq \theta_2$, implies $\mathcal{N}_x(\theta_1) \preceq \mathcal{N}_x(\theta_2)$ and $\mathcal{N}_y(\theta_1) \preceq \mathcal{N}_y(\theta_2)$: implicit is the assumption that the precedence relation $\preceq$, defined on Θ is carried over to the collection of neighbourhoods for signals x and y.

In most filtering operations, elements of Θ ($\equiv \Xi$) are given connotations of *time* and/or *spatial co-ordinates* and consequently, the precedence relation $\preceq$ has the natural interpretation of historicity. *Anticipatory systems* (*cf*, **Rosen**, 1985) are those filters whose neighborhood structures defy the stricture of causality: in such systems it is common to interpret the processing as being influenced by signal assignments of an unvisited future. It is interesting to note that signals are identifiable with filters (processors), thereby reducing the distinction between signals and (signal processing) systems. Filters with elements of randomness, are termed *stochastic processes*, when the signal definition domain is one-dimensional and *random fields* otherwise (*ie*, higher-dimensional signal definition domain).

Transforms, in signal processing, are defined to be situations of processing established between non-identical domains of signal definition (*ie*, $\Xi \not\equiv \Theta$) and it is common to force $\mathcal{N}_y(\theta) \equiv \emptyset$ for all $\theta \in \Theta$. Classical transforms are characterized by $\mathcal{N}_x(\theta) = \Xi$ for all $\theta \in \Theta$, while the approach in window transforms (including wavelet transforms) is to seek out a neighbourhood structure of the form $\forall\, \theta_1, \theta_2 \in \Theta$, $\theta_1 \neq \theta_2$, implies $\mathcal{N}_x(\theta_1) \mathbin{\Delta} \mathcal{N}_x(\theta_2) \neq \emptyset$. Usually, Ξ is associated with time and/or spatial-extent and Θ with frequency, or repeatability of (specific) relative organization of assignments and the signal y is, generally, termed the *spectrum* of the signal x. A reversal of interpretations is used in *inverse transforms*.

In the representation of signals and linear (shift invariant) processors, transforms play an important role. The transform (or spectral) approach simplifies an operation of convolution to point-wise multiplication (with non-window transforms like Fourier transform, Laplace transform, or their variants): this feature allows the processor functionality to be completely characterized in terms of its *impulse response*. Finiteness of the impulse response is used to characterize (linear) processors, specifically filters.

Windowed transforms, essentially (conventional) transforms applied to local signal sections, introduce a spectral dependence on shifts (of windows, described through the measurement function ϕ) in the signal definition domain. This dependence motivates window transforms to be viewed as members of the general class of *Spatial-Spectral processors*, of which time-frequency processors (*cf*, **Cohen**, 1989) and processors involving cojoint spatial/spatial-frequency (energy) distributions (*cf*, **Wechsler**, 1990) are important special cases. Joint characterization of processors, on the signal definition and spectral domains have been immensely useful in processing signals with non-stationarity in statistics and also when signals are to be subjected, as in Wavelet Transforms, to non-uniform processing, possibly without shift-invariance

Spatial-Spectral processors are represented in the formalism of Equation 2.1 (*p.* 39) with the following refinement of symbols. Identify with $\varTheta$, two domains $\tilde{\varTheta}$ and $\tilde{\varXi}$, such that $\varTheta = \tilde{\varXi} \times \tilde{\varTheta}$, where, $\tilde{\varXi}$ is derived from $\varXi$ and $\tilde{\varTheta}$ is given spectral connotations like frequency, or scale (Wavelet Transforms) The variable θ is identified with the ordered 2-tuple $(\tilde{\xi},\tilde{\theta})$ with $\tilde{\xi} \in \tilde{\varXi}$ and $\tilde{\theta} \in \tilde{\varTheta}$. Note that the neighbourhoods ($\mathcal{N}$) and assignments ($\mathfrak{a}$) related to signals x and y are now indexed by $\tilde{\xi}$ and $\tilde{\theta}$. Rephrase the measurement functions ϕ and ψ as

$$\phi(\mathfrak{a}_x(\theta)\,,\theta) \equiv \phi(\mathfrak{a}_x(\tilde{\xi},\tilde{\theta})\,,\tilde{\xi},\tilde{\theta}) = \tilde{\phi}(\mathfrak{w}_x(\tilde{\xi},\tilde{\theta})\,,\mathfrak{a}_x(\tilde{\xi},\tilde{\theta})\,,\tilde{\xi},\tilde{\theta})\,,$$
$$\psi(\mathfrak{a}_y(\theta)\,,\theta) \equiv \psi(\mathfrak{a}_y(\tilde{\xi},\tilde{\theta})\,,\tilde{\xi},\tilde{\theta}) = \tilde{\psi}(\mathfrak{w}_y(\tilde{\xi},\tilde{\theta})\,,\mathfrak{a}_y(\tilde{\xi},\tilde{\theta})\,,\tilde{\xi},\tilde{\theta})\,,$$

with appropriate choice of component functions $\tilde{\phi}$, $\tilde{\psi}$, $\mathfrak{w}_x$ and $\mathfrak{w}_y$. In the above symbolization, $\mathfrak{w}_x(\tilde{\xi},\tilde{\theta})$ and $\mathfrak{w}_y(\tilde{\xi},\tilde{\theta})$ denote the window functions operating on x and y respectively. In window transforms, it is common to find $\tilde{\varXi} \equiv \varXi$, and members of $\tilde{\varXi}$ signify the amount of translations the (weighting) windows are subjected to.

Gabor Transforms and Wavelet Transforms are the most commonly used window transforms in signal processing. Noting that $\mathcal{N}_y(\tilde{\xi},\tilde{\theta}) \triangleq \mathcal{N}_y(\theta) \equiv \emptyset$ for all $(\tilde{\xi},\tilde{\theta}) \in \tilde{\varXi} \times \tilde{\varTheta}$, in the case of (window) transforms, the processor is specified by the simpler abstraction

$$\forall \left(\tilde{\xi},\tilde{\theta}\right) \in \tilde{\varXi} \times \tilde{\varTheta} \quad y(\tilde{\xi},\tilde{\theta}) = \tilde{\phi}(\mathfrak{w}(\tilde{\xi},\tilde{\theta})\,,\mathfrak{a}(\tilde{\xi},\tilde{\theta})\,,\tilde{\xi},\tilde{\theta})\,,$$

wherein recursive (self-referential) dependence of y has been suppressed. Gabor transforms are characterized by

$$\begin{aligned}
(\mathfrak{a}(\tilde{\xi},\tilde{\theta}))(\xi) &= e^{-\imath\tilde{\theta}\xi}x(\xi)\ \forall\xi\in\Xi,\\
(\mathfrak{w}(\tilde{\xi},\tilde{\theta}))(\xi) &= \frac{1}{2\sqrt{\pi\varsigma}}e^{-(\xi-\tilde{\xi})^2/4\varsigma^2}\ \forall\xi\in\Xi,\\
\tilde{\phi}(\mathfrak{w}(\tilde{\xi},\tilde{\theta}),\mathfrak{a}(\tilde{\xi},\tilde{\theta}),\tilde{\xi},\tilde{\theta}) &= \int_{\Xi} d\mu(\xi)\,(\mathfrak{a}(\tilde{\xi},\tilde{\theta}))(\xi)\,(\mathfrak{w}(\tilde{\xi},\tilde{\theta}))(\xi)\\
&\qquad \forall\left(\tilde{\xi},\tilde{\theta}\right)\in\tilde{\Xi}\times\tilde{[}\Theta],
\end{aligned}$$

where, $\imath^2 = -1$ and $\varsigma > 0$ is a constant uninfluenced by choices in translational ($\tilde{\xi}$) or spectral ($\tilde{\theta}$) parameters. (Note that $\mathfrak{w}(\tilde{\xi},\tilde{\theta}) \equiv 1$ allows Fourier transforms to be computed.)

In contrast, Wavelet Transforms have the first two of the above expressions as

$$\begin{aligned}
(\mathfrak{a}(\tilde{\xi},\tilde{\theta}))(\xi) &= x(\xi)\ \forall\xi\in\Xi,\\
(\mathfrak{w}(\tilde{\xi},\tilde{\theta}))(\xi) &= |\tilde{\theta}|^{-1/2}\mathfrak{b}(\frac{\xi-\tilde{\xi}}{\tilde{\theta}})\ \forall\xi\in\Xi,
\end{aligned}$$

where $\mathfrak{b}$ denotes a *basic wavelet*[11] window. The basic wavelet window is commonly obtained from a scaling function φ as the solution of the

[11] In the literature, basic wavelets are denoted as ψ and scaling functions from which the wavelets are derived, by scaling and shifting, are denoted as ϕ. As these symbols have already been used to denote abstract signal processing steps, I have opted to recode the notations to preserve conceptual clarity in the symbols.

difference equations[12]

$$\begin{aligned}\varphi(\xi) &= \sum_{i=-\infty}^{+\infty} p_i\varphi(2\xi - i),\\ \psi(\xi) &= \sum_{i=-\infty}^{+\infty} q_i\varphi(2\xi - i),\end{aligned}$$

the ℓ^2 sequences p and q being called *two scale* sequences.

As localized evaluation is the key idea of windowing, transforms based on wavelets (subjected to scaling and translation) are considered superior when the extent of localization is to adapt in accordance with signal variation over the domain of definition. (In Gabor transforms, localization of signal evaluation is invariant in the sense that no scale parameter is incorporated.) Window transforms, by virtue of localization in signal assessment, have been used in time-frequency (spatial/spatial-frequency) analysis wherein the focus is to trace (detect) variations in signal characteristics, typically expressed through terms having spectral connotation, as a function of signal evolution in time (space). An important point to note is that in window transforms, localization normally is effective in the signal definition as well as spectral domains and is unbiased towards the dimension of these spaces, *ie*, localization is operative on all the basis vectors describing the signal.

[12] Similar to two scale relationships, general n scale relationships ($n = 2, 3, \ldots$), while conceivable, have not yet become popular and hence the general difference equations are not indicated.

2.2 Artificial Neural Networks: A preparatory review

Research in artificial neural networks, in a span of a little over a half century, has had the issues related to the modeling of (human) perceptual activities, specifically the information processing sought to be carried out by the brain, as a continuing area of major interest: one of the motivations for such a focus is to eventually enable an automated expression of intelligence. In this focus, the key problems addressed have been the identification (and isolation) of plausible structures capable of information processing, typically categorization, generalization and estimation, especially when the patterns presented (for processing) are noisy and/or incomplete; information retention with associated issues of retrieval and recall, in particular models accounting for short and long term memories; and automated mechanism(s) of incorporating available knowledge into suggested structures of information processing and/or storage: this latter problem is known as learning.

The underlying paradigm in artificial neural networks is to realize all of the above (perceptual) capabilities through an interconnected ensemble of basic processing units, these units are themselves not expected to exhibit any of the desired functional traits: such an operational characteristic encourages the view that neural networks (of the natural as well as artificial kinds) allow for an 'emergence' of computational functionality through the framework of interconnections – the notion of generalization is not unrelated to that of emergence. Information retention, retrieval and recall has been sought through interconnected hierarchical dynamical systems, whereas information processing has been accounted for in a variety of architectures, each attempting to recreate a distinct facet of biological information processing.

An unbroken tradition of research in artificial neural networks has been the parametrically selectable processing character of the constituent units: the response of a node to information incident on the input channels (*ie*, dendritic arborescence in real world neurons) is influenced by the specific weightages (parameters) associated with the channels and the parameters are tuned (selected) in accordance with the specific (desired) knowledge base to be represented, or learnt.[13] In all of the presentations of neural network research, (information bearing) patterns of activity presented to the processing nodes, weightages associated with information incident on input channels and the response of the basic information processing units are encoded using real numbers (at times integers), this encoding is used for a conjoint representation of coordinates and objects in the perceived reality around us. Each processing node is then, a multivariate (real) scalar valued (real) function and it is common to expect, from considerations of categorization, that this function is of a nonlinear nature.

[13] The criticism by (**Edelman**, 1987) of information processing models accounting for intelligence rests on the essential problems of the *a priori* status to information and agental (homuncular) status to information association central to parametric formulations of information processing.

Learning, or the automatic association with processing nodes, of connection strengths capable of representing the desired knowledge base, has presented the most challenging of problems in neural network investigations. Nearly all formulations of the learning problem seek to search an optimal choice of parameters in an appropriate space, the criterion of optimality is generally chosen from considerations of enabling the search to be approached as a variant of gradient-descent

The nonlinear nature of the functional relationship between inputs and output of the processing node compounds the search problem by imposing a non-unimodality in the (appropriately signed) objective function: a consequence of this lack of unimodality is that solutions to the search problem are generally locally optimal, instead of the desired global optimum. Retrieval and recall in dynamic, or recurrent, neural networks being similar to the problem of learning, local optimal solutions spell disaster due to incomplete pattern reconstruction, *ie*, inaccurate recall with no assurance of repetitive attempts of similar recall resulting in similar responses. Issues of generalization, when included along with representation (*ie*, learning), further complicates the situation, rendering the learning problem intractable.

Success, even though limited, in modeling (human) cognitive abilities, have triggered a fresh wave of interest in artificial neural networks. In this section I will review the nature of research in artificial neural networks, principally from the point of view of neuro-science, to facilitate a review of neural signal processing, one of the key activities under the heading of neuro-engineering. As the focus, in artificial neural networks, has really been on models accounting for perception, significant attention has not been given, in the present endeavors, to the issue of data/knowledge validity, as is to be expected in any modeling endeavor, typically using statistical approaches.[14]

Neurons and their Models

Isolated real world neurons,[15] used as the functional basis of artificial neural networks, are formally modeled by expressions having the general form

$$\begin{aligned} \frac{\partial}{\partial t}\{\eta(\underline{x},t)\} &= -a(\eta(\underline{x},t))\left(b(\eta(\underline{x},t)) - \sum_{i=1}^{n} s_i x_i\right), \qquad (2.10a) \\ y(\underline{x},t) &= \sigma(\eta(\underline{x},t)\,,t)\,; \qquad (2.10b) \end{aligned}$$

the earlier of these equations is equivalent to the statement

$$\dot{\eta}(\underline{x},t) = -a(\eta(\underline{x},t))\,(b(\eta(\underline{x},t)) - \underline{s}\cdot\underline{x})\,, \quad \dot{} = \frac{\partial}{\partial t},$$

[14] Thus, detection of outliers in the training data, robustness of the neural procedures in information processing and robustness of learning algorithms have not yet been considered sufficiently important, though, as the paradigm of neural networks is increasingly attracting the attention of statisticians, these issues will tend to dominate future discourses on neural networks. Algorithmic issues, principally that of complexity, currently addressed sporadically, are bound to highlight, in subsequent times, the characteristics of computation with artificial neural networks.

[15] In this discussion, I will dwell only on the mathematical models suggested for (and inspired from) biological neurons. Anatomical and physiological basis of models for neurons and their interconnected ensembles can be found in the biologically grounded discussions provided by **Churchland** (1986), **McClelland**, **Rumelhart**, *et al* (1986a), **Peretto** (1992) and **Churchland & Sejnowski** (1994).

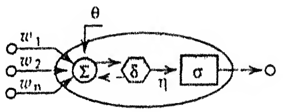


Figure 2.2· Symbol for an isolated neuron

and the latter is sufficiently general to allow incorporation of *refractory tımes* (*cf*, **Peretto**, 1992) in the neural response. Neurons with the above (internal) dynamics are said to be of the ***Cohen & Grossberg (1983)*** type. (See, **Kosko**, 1992a) This is a deterministic model for the dynamics within neurons (*ie*, elementary processors of a neural network). Figure 2.2 depicts the above model in graphical symbols and this symbolism will be used in the sequel to denote isolated neurons.

Additive and shunting models are the most important special cases of the above dynamical model of a neuron. When a is a constant and the function b is rectilinear[16] in its argument the neuron is said to have *additive* dynamics (**Kosko**, 1992a). The model of additive dynamics, also termed in the literature as *brain-state-in-a-box* (**Anderson**, 1983) and *conductance model*, is typically (an equivalent) of the following form.

$$C\frac{d}{dt}\{\eta(\underline{x},t)\} + \frac{\eta(\underline{x},t))}{R} \quad = \quad \sum_{i=1}^{n} \frac{x_i}{R_i} - I$$

[16] A functıon $f\colon \Re \rightarrow \Re$ is rectilinear in its argument, say x, if f has the form $f(x) = mx + c$, for some $m, c \in \Re$. This, incidentally, is not the same as linearity when $c \not\equiv 0$.

$$\begin{aligned} \equiv \tau\dot{\eta}(\underline{x}, t) + \eta(\underline{x}, t) &= \sum_{i=1}^{n} w_i x_i - \theta, & (2.11a) \\ y(\underline{x}, t) &= \sigma(\eta(\underline{x}, t)), & (2.11b) \end{aligned}$$

where, $a(\eta(\underline{x}, t)) \equiv C^{-1}$, $b(\eta(\underline{x}, t)) = R^{-1}\eta(\underline{x}, t) - I$, and $s_i = R_i^{-1}$, C, R, R_i and I being constants; the refractory time in neural response has been ignored and will not be considered in the rest of the discussion.

If, in Equation 2.10 (*p.* 57), the function a is rectilinear in its argument and b is nonlinear, *shunting* or *multiplicative* activation dynamics results in the neuron which represents a special case of the *Hodgkin-Huxley membrane equation* (*cf*, **Hodgkin & Huxley**, 1952; **Grossberg**, 1982; 1988; **Cohen & Grossberg**, 1987; **Kosko**, 1992a). This model exhibits saturation (with increasing pattern intensity) if the function b is a constant (*ie*, a trivial nonlinear function).

In this discussion, the following notations hold.[17]

η := potential accumulated on the membrane of a neuron (*ie*, neuron state, also termed as post synaptic potential), $\eta \in \Re$, $\Re$ is the real number field.

x_i := activity (input) on (dendritic) channel i, $i = 1, 2, \ldots n$, n being the number of channels, and $x_i \in \Re$.

t := (independent) variable denoting the progression of time, $t \in [0, \infty]$.

[17] Note that θ has been used again in a sense different from that in the previous section. In this section and the rest of the thesis, θ will be used to mean threshold and/or bias associated with a neuron: the distinction will be clear from the context.

a := an abstract amplification function indicating the mechanism of modulation (decay) of the membrane potential η, a: $\Re \rightarrow \Re$, with the restriction that a takes non-negative values for reasons of stability (**Kosko**, 1992a).

b := an abstract translation function specifying the extent of state translation in the dynamics of the membrane potential η, b: $\Re \rightarrow \Re$.

s_i := interconnection strength, or synaptic efficacy, of channel i, $i = 1, 2, \ldots n$, $s_i \in \Re$

y := action (response) of the neuron, physiologically associated with the frequency of axonal spike generation, $y \in [\zeta_-, \zeta_+] \subset \Re$ for neurons with continuous valued outputs with appropriate values for ζ_- and ζ_+, or $y \in \{\zeta_0, \zeta_1, \ldots \zeta_c\}$, with *a priori* values $\zeta_j \in \Re$, $j = 0, 1, \ldots c$, $c = 1, 2, \ldots$ being (one less than) the number of categories, for neurons with discrete valued outputs.

σ := activation function mapping the membrane potential η to the response (axonal spike frequency) y, generally using a non-linear method (possibly with a provision for refractory time), σ: $\Re \rightarrow [\zeta_-, \zeta_+]$ for continuous valued neurons and σ: $\Re \rightarrow \{\zeta_0, \zeta_1, \ldots \zeta_c\}$ for discrete valued neurons.

C := membrane capacitance (constant amplification in conductance model), $C > 0$.

R := membrane (leakage) resistance (linear translation in conductance model), $0 < R < \infty$.

R_i^{-1} := conductance of channel i (connection strength in conductance model), $0 < |R_i^{-1}| < \infty$, $i = 1, 2, \ldots n$.

I := current applied externally (static translation in conductance model), $I \in \Re$.

$\tau = RC$:= membrane charge-discharge time constant, $0 < \tau < \infty$.
$w_i = \frac{R}{R_i}$:= weighting value associated with channel i, $w_i \in \Re$, $i = 1, 2, \quad n$
$\theta = RI$ = threshold of firing, $\theta \in \Re$

Note that the abstract translation function b serves to provide a generalization of thresholding.

The activation function σ is nonlinear, in general, to allow for the decision, as a function of the membrane potential, to be non-trivial.[18] A general requirement, from considerations of categorization and decision-making, is that the activation function be capable of inducing a discrimination on the membrane potential η. Common forms for the activation function are

$$\text{hardlimiter} := \quad \sigma_h(\xi) = \begin{cases} \zeta_+ & \text{if } \xi \geq 0, \\ \zeta_- & \text{otherwise,} \end{cases} \tag{2.12a}$$

$$\text{sigmoid} = \quad \sigma_s(\xi) = (\zeta_+ - \zeta_-)\,\frac{(1 + tanh(\xi))}{2} + \zeta_-, \tag{2.12b}$$

$$\equiv \begin{cases} \frac{(1+tanh(\xi))}{2} = \frac{1}{1+e^{-2\xi}} & \\ & \text{if } [\zeta_-, \zeta_+] = [0, 1], \\ tanh(\xi) & \text{if } [\zeta_-, \zeta_+] = [-1, 1], \end{cases} \tag{2.12c}$$

where, $\zeta_-, \zeta_+ \in \Re$, $\zeta_- < \zeta_+$. Monotonicity in the activation function is considered important from the point of view of biological models of neurons. However, in the context of neural decision making, this

[18] In real world neurons, the activation function is sought to establish the dependence of the frequency of axonal spike generation on the membrane potential. Higher the frequency, greater is the level of activity in the neuron.

condition is, at times, relaxed and discrimination with non-monotonic activation function is typically based on functions of the form

$$\sigma_g(\xi) = (\zeta_+ - \zeta_-)\, exp(-\frac{\xi^2}{2}) + \zeta_-, \qquad (2.13)$$

which resemble Gaussian functions[19] normalized to have unit variance. As monotonicity in activation functions helps preserve partial orderings in the input space, functions used for discrimination are, in general, *piece-wise monotonic*. Though this term is obviously redundant, its usage explicitly specifies localized monotonicity, which essentially points to localized preservation of input space partial orderings.

It is common to consider the transients due to input transitions as decaying rapidly and thus it is of interest to consider the steady state response of a neuron. The steady state neural model with additive dynamics is then of the typical form

$$\begin{aligned} \eta(\underline{x}) &\triangleq \lim_{t\to\infty} \eta(\underline{x},t) &&= \sum_{i=1}^{n} w_i x_i - \theta, \text{ as } \lim_{t\to\infty} \dot{\eta}(\underline{x},t) = 0, && (2.14a)\\ y(\underline{x}) &\triangleq \lim_{t\to\infty} y(\underline{x},t) &&\equiv \lim_{t\to\infty} \sigma(\eta(\underline{x},t)) = \sigma(\eta(\underline{x})), && (2.14b)\end{aligned}$$

which resembles the formal model originally proposed by **McCulloch & Pitts** (1943), adapted, later on, by subsequent investigators. In the case of multiplicative dynamics with the abstract amplification function

[19] A variation of this scheme, known as *radial basis function* networks are discussed in the literature. These networks use quadratic, rather than linear, discriminants evaluated as a norm – generally Euclidean – of the difference between the input vector $\underline{x}$ and appropriately chosen vectors (in the same space as $\underline{x}$), say $\underline{x}_i$, $i = 1, 2, \ldots, N$, for some given N Discrimination is non-monotonic and is through functions with even symmetry, typically Gaussian, operating on the quadratic discriminants.

being $a(\eta) = A_1\eta$ and the abstract translation function taking a form $b(\eta) = B_{-1}\eta^{-1} + B_0$, and the variation, if any, in the input $\underline{x}$ being *reasonably* slower than the decay of transients, the steady state model of the neuron is

$$\eta(\underline{x}) \triangleq \lim_{t\to\infty} \eta(\underline{x}, t) = \frac{-B_{-1}}{B_0 + \underline{s}\,\underline{x}}, \tag{2.15a}$$

$$y(\underline{x}) \triangleq \lim_{t\to\infty} y(\underline{x}, t) \equiv \lim_{t\to\infty} \sigma(\eta(\underline{x}, t)) = \sigma(\eta(\underline{x}))\,. \tag{2.15b}$$

The steady state versions of the formal models amply suggest that the neural state variable η (membrane, or post-synaptic potential) is influenced by $\underline{x}$, the pattern incident on the input channels, through a projection along a vector of interconnection strengths. Topologically, this projection implies that with bivalent[20] neurons (as expected with hardlimiter activation function) the space (set) of input patterns is partitioned into two distinct regions, the dichotomy being decided by a linear manifold, *ie*, a (hyper)-plane: the input pattern space is then said to be *linearly separated* (**Cover**, 1965; **Minsky & Papert**, 1969; **Lippmann**, 1987; **Matheus & Hohensee**, 1987).

Boolean functions being bivalent, **McCulloch & Pitts** (1943) and similarly **Cover** (1965) and **Hurst** (1971) working in threshold logic, demonstrated that bivalent neurons are capable of realizing Boolean functions, thereby providing a framework for representing formulae of the propositional calculus, an idea that inspired the early design of logic gates and consequently digital computers. **Rosenblatt** (1961), in his

[20] Bivalent neurons are also known in the literature as ’binary’ neurons.

study of perceptrons,[21] provided an algorithm, with proven and desirable convergence characteristics, for an automated specification of the strengths of modifiable connections associated with the (single layer of) decision units (processing the outcomes of [static] predicates operating on the input pattern) given the required (bivalent) mapping to be imposed on chosen patterns. this scheme, involving neural implementations of predicates and decision units, together with the perceptron learning algorithm, was suggested to be a model for perception, learnable from examples, in biological systems.

Minsky & Papert (1969), however, established that single (bivalent) neurons by virtue of the linear separation induced on the space of input patterns, are, in general, incapable of representing all Boolean functions and showed that the parity function (also known as XOR) is one of many Boolean functions which demand a separation different from that provided by linear manifolds. This limitation had already triggered an inquiry into the representational capacity of networks of neurons under the heading of multi-layer perceptrons (*cf*, **Rosenblatt**, 1961), however, inadequate[22] automation in the specification of interconnection strengths in multiple layers of neurons discouraged the deployment of the neural processing alternative, till the usage of (stochastic) gradient descent in the learning of interconnection strengths.

[21] See the description of perceptrons, by Rumelhart and McClelland, quoted earlier.

[22] The reasoning provided by **Minsky & Papert** (1969) has, in the literature, been attributed, possibly inaccurately, to an onset of near dormancy (between 1970 and 1985) in neural network research. However, most of the architectural and procedural, innovations in neural networks have been conducted in exactly this period!

In the case of neurons with bivalency, as provided by the hardlimiter activation function, additive and multiplicative dynamics, specially in the steady state, are really equivalent, *ie*, every dichotomy under one scheme is realizable under the other. This equivalence is, however, not noticed with continuous activation functions like sigmoid and radial basis. A careful reflection on the reciprocal dependency relationship in the steady-state version of the formal model of shunting dynamics, as compared to a direct dependence in additive dynamics, reveals that multiplicative dynamics allows the geometry provided by additive dynamics to be inverted. Figure 2.3 compares the discrimination provided, in steady state, by hardlimiter, sigmoid and radial basis functions under additive dynamics.

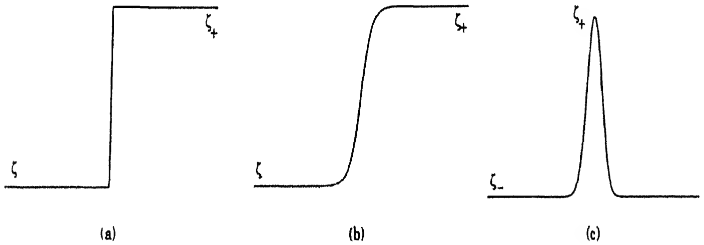


Figure 2.3: Comparison of discrimination, in steady state, under additive and multiplicative dynamics

Networks of Neurons

Single neurons, especially of the linear separating type, are not powerful enough to realize (or approximate) all the decision (or classification) functions of interest (*cf*, **Minsky & Papert**, 1969; **Lippmann**, 1987), and consequently, networks of neurons have been explored in the literature. A neural network, is essentially an interconnected ensemble of local dynamical systems (also oscillators) and the resulting dynamics, if any, of such a system are due to the structure of inter-neuron interconnection in addition to dynamics supported by individual neurons.

One of the basic tenets of neuroscience is that complex behaviors such as sensory perception or motor control, exhibited by the brain, arise from the interconnection of neurons into networks or circuits The interconnection structure defines the network architecture and current classification of neural networks rests on this principle. Formally, a neural network is specified by the following.

$$\begin{aligned}
\dot{\eta}^{(1)}_{j^{(1)}}(\underline{x}, t) &= a^{(1)}_{j^{(1)}}(\eta^{(1)}_{j^{(1)}}(\underline{x}, t)) \Bigg[b^{(1)}_{j^{(1)}}(\eta^{(1)}_{j^{(1)}}(\underline{x}, t)) + \underline{s}^{(1)}_{j^{(1)}} \cdot \underline{x} \\
&\qquad + \sum_{i=1}^{m_{j^{(1)}}} \epsilon^{(1)}_{j^{(1)}\,i}\, y^{(1)}_i(\underline{x}, t - \tau^{(1)}_{r_{j^{(1)}}\,i}) \Bigg], \\
y^{(1)}_{j^{(1)}}(\underline{x}, t) &= \sigma^{(1)}_{j^{(1)}}(\eta^{(1)}_{j^{(1)}} \underline{x}, t)), \\
&\qquad j^{(1)} = 1, 2, . \ . m_1, \\
&\qquad \text{for some } m_1 = 1, 2, \ldots, \qquad (2.16a)
\end{aligned}$$

$$\begin{aligned}
\overset{\bullet}{\eta}{}^{(\ell)}_{j^{(\ell)}}(\underline{x},t) &= a^{(\ell)}_{j^{(\ell)}}(\eta^{(\ell)}_{j^{(\ell)}}(\underline{x},t)) \Bigg[b^{(\ell)}_{j^{(\ell)}}(\eta^{(\ell)}_{j^{(\ell)}}(\underline{x},t)) \\
&\quad + \sum_{i_f=1}^{m_{j^{(\ell-1)}}} s^{(\ell)}_{j^{(\ell)}\, i_f} y^{(\ell-1)}_{i_f}(\underline{x}, t - \tau^{(\ell)}_{f_{j^{(\ell)}}\, i_f}) \\
&\quad + \sum_{i_r=1}^{m_{j^{(\ell)}}} \epsilon^{(\ell)}_{j^{(\ell)}\, i_r} y^{(\ell)}_{i_r}(\underline{x}, t - \tau^{(\ell)}_{r_{j^{(\ell)}}\, i_r}) \Bigg], \\
y^{(\ell)}_{j^{(\ell)}}(\underline{x},t) &= \sigma^{(\ell)}_{j^{(\ell)}}(\eta^{(\ell)}_{j^{(\ell)}}(\underline{x},t)), \\
&\qquad j^{(\ell)} = 1,2,\ldots m_\ell, \quad \text{for some} \quad m_\ell = 1,2,\ldots, \\
&\qquad \ell = 2,3,\ldots L, \text{for some} L = 1,2,\ \ldots \qquad (2.16b)
\end{aligned}$$

Note that the above specification does not disagree with the definitions suggested by DARPA (**Sage & Withers**, 1990) and **Hecht-Nielsen** (1990) Layers, in neural network related investigations of physicists, have also been identified with *fields*[23] (*cf*, **Amari**, 1983; **Sompolinsky**, 1987; 1988; **Hopfield**, 1982; **Grossberg**, 1982): this terminology allows neural networks to be discussed in field theoretic terms. The above specification is really not very general, as it precludes feedback between non-adjacent layers, though, keeping with the current tradition in neural networks, incorporation of such feedback would follow the same principles guiding intra-layer interactions.

Layering, in the specification of neural networks, admits external stimuli only into a specific layer, termed *input layer* and protects (dis-

[23] When a quantity $\psi(x)$ is defined at every point x in a certain region of a space physicists say that a field of the quantity ψ is given (**Itô**, 1987).

allows) nodes in other layers from being directly influenced by external stimuli. Similarly, only a specific layer, termed *output layer*, is allowed to inform the external world of the processing accomplished by the network, other layers play the role of providing a mechanism for intermediate (and internal) computations and are not allowed to directly influence the external environment. Layers not in direct interaction with the external environment are labelled as being *hidden*.

In the neural network literature, a general disagreement is evident in assigning the role of input layers. Consequently, a confusion and inconsistency, pervades in the numbering of layers and the number of layers to be ascribed to a (layered) network of neurons, though a consensus, borne out of our natural sense of ordering, prevails in that the numbers assigned to layers increase as the degree of (inter-neural) association, in a sense equating the layer number with depth of information processing incorporated on external inputs (stimuli).

Early research, attempting to model (sensory) perception and motor control through neural information processing mechanisms, (*eg*, **Rosenblatt**, 1958; **Albus**, 1975) has shown an inclination to devote the input layer to gather external stimuli and the role of this layer in information processing has been to fan-out the collected stimuli to appropriate decision units (predicates) in the network of neurons. In contrast, research wherein neural networks are viewed as a computational substrate (*cf*, **Lippmann**, 1987), has had the input layer participate in decision making and the role of distributing external stimuli to appropriate channels (inputs) of decision units has been merged with the functionality of variable strength interconnections. The latter view is better suited to discussions of signal processing with neural networks and has been incorporated in the specification of networks of neurons.

The following notations hold in the preceding equations.

L .= number of layers in the network.

m_ℓ ·= number of processing nodes (ie, neurons) in layer ℓ, $\ell = 1, 2, \ . \ L$.

$\underline{s}^{(\ell)}_{j^{(\ell)}}$.= feed-through synaptic efficacies (ie, inter-layer interconnection strengths) for processing node $j^{(\ell)}$ in layer ℓ, $j^{(\ell)} = 1, 2, \ . \ m_\ell$, $\ell = 1, 2, \quad L$.

$\underline{\epsilon}^{(\ell)}_{j^{(\ell)}}$:= feed-back (recurrent) synaptic efficacies (ie, intra-layer interconnection strengths) for processing node $j^{(\ell)}$ in layer ℓ, $j^{(\ell)} = 1, 2, . \ m_\ell$, $\ell = 1, 2, \quad L$.

$\tau^{(\ell)}_{f_{j^{(\ell)}, i_f}}$.= propagation (and refractory) delay in the feed-through path from processing node i_f of layer $(\ell - 1)$ to processing node $j^{(\ell)}$ in layer ℓ, $i_f = 1, 2, \quad m_{\ell-1}$, $j^{(\ell)} = 1, 2, \ . \ m_\ell$, $\ell = 1, 2, . \ L$.

$\tau^{(\ell)}_{r_{j^{(\ell)}, i_r}}$:= propagation (and refractory) delay in the feed-back path from processing node i_r to processing node $j^{(\ell)}$, both in layer ℓ, $i_r = 1, 2, . \ m_\ell$, $j^{(\ell)} = 1, 2, . \ . m_\ell$, $\ell = 1, 2, \ldots L$.

For convenience, delay terms related to external stimuli (ie, pattern $\underline{x}$) have been ignored: this would not (appreciably) alter the usefulness of the discussion in the not-so-unrealistic case of transitions in (externally applied) inputs being less frequent than the change in the state variables due to internal dynamics. If the transients in neural response due to input transitions settle rapidly, and thereby can be ignored and in addition, all feed-back delays are of unit duration (and feed-through delays are null), then the above expressions for an L-layered neural network, with additive dynamics in the neurons, have the following simpler form,[24] in steady state.

$$\begin{aligned}
\eta^{(1)}_{j^{(1)}}(\underline{x},\nu) &= \underline{w}^{(1)}_{j^{(1)}}\,\underline{x} + \underline{c}^{(1)}_{j^{(1)}}\;\; \underline{y}^{(1)}(\underline{x},\nu-1) - \theta^{(1)}_{j^{(1)}}, \\
y^{(1)}_{j^{(1)}}(\underline{x},\nu) &= \sigma^{(1)}_{j^{(1)}}(\eta^{(1)}_{j^{(1)}}(\underline{x},\nu)), \\
&\qquad\qquad j^{(1)} = 1, 2, \quad . m_1, \\
&\qquad\qquad \text{for some } m_1 = 1, 2, \quad . \qquad\qquad (2.17a) \\
\eta^{(\ell)}_{j^{(\ell)}}(\underline{x},\nu) &= \underline{w}^{(\ell)}_{j^{(\ell)}}\;\; \underline{y}^{(\ell-1)}(\underline{x},\nu) + \underline{c}^{(\ell)}_{j^{(\ell)}}\;\; \underline{y}^{(\ell)}(\underline{x},\nu-1) - \theta^{(\ell)}_{j^{(\ell)}}, \\
y^{(\ell)}_{j^{(\ell)}}(\underline{x},\nu) &= \sigma^{(\ell)}_{j^{(\ell)}}(\eta^{(\ell)}_{j^{(\ell)}}(\underline{x},\nu)), \\
&\qquad\qquad j^{(\ell)} = 1, 2, \quad . m_\ell, \text{ for some } m_\ell = 1, 2, \ldots, \\
&\qquad\qquad \ell = 1, 2, .. \; L, \qquad\qquad (2.17b)
\end{aligned}$$

where ν is the discrete time travel index.

From the above simplification it is easy to see that each layer of a network of bivalent neurons, by virtue of a conjunctive logic naturally operating on the decision regions represented by individual nodes, partitions the space of patterns seen by the processing nodes of that layer into numerous regions, each manifesting as an appropriate subset of

[24] This form is popular in the computational explorations of neural networks and suggests the possibility of a unification of neural networks with other dynamical systems like Cellular Automata, (Universal) Turing Machines, Discrete Event Systems, *etc*, under the haeding of *function fields over lattice index spaces*: the functions in the field are, in general, between (appropriately chosen) manifolds.

the desired partition in the space of (externally) applied input patterns. Though no thorough characterization of the nature of decision regions induced by multi-layered neural networks is available, **Lippmann**, 1987 has argued out, geometrically, that separation of the input pattern space for networks with two or more layers is, in general, through a nonlinear manifold and the nature of nonlinearity has been described in terms of convexity,[25] and closedness, of the decision regions induced in a space of patterns isomorphic to the Euclidean space $\Re^n$ of dimensionality n, the number of inputs to the nodes of the input layer.

Single layer networks (of bi-valent neurons) are shown to partition the input space into two convex regions, at least one being non-null and all non-null regions being open (*ie*, unbounded). Two-layer networks partition the input space into two regions, at least one being non-null and no more than one of the non-null regions being non-convex and no more than one convex region being closed (bounded) with a single (connected) component. A network involving a cascade of three-layers of (bi-valent) neurons partitions the input space into two regions, at least one being non-null, all non-null regions allowed to be non-convex and no more than one non-convex region being closed (bounded), however, this closed region is allowed to have multiple (connected) components. It is commonly presumed, though not rigorously proved, that three layers are adequate to realize all binary partitions on Euclidean spaces.

[25] The nature of decision regions induced by (bivalent) neural networks, encourages the view that neural network approach might be suitable to handle convex programming and be of immense use in related optimization and search problems

The tradition of neuroscience being physicalist (and reductionist), it is common to find the network structures investigated to be of a homogeneous (also regular) kind, *ie*, all nodes of all layers are similar, if not identical Neural network taxonomy has been attempted only on such regular structures and the basis for the taxonomy is provided by the nature of neural dynamics, inclusive of the type of activation function and specification of inter-layer and intra-layer interconnections, which includes characterization of the time-translation, generally delays, involved in information propagation along these interconnections

If all the intra-layer interconnection strengths (*ie*, ϵ terms with appropriate indices) are null (zero), then the resulting network structure is termed a *feed-forward* network, else the network is said to be *recurrent* Dynamics in the network response is due to isolated contribution of processor (internal) dynamics, as in feed-forward networks (*eg*, perceptrons), or intra-layer interactions, as in (auto)-associative networks,[26] or a combined influence of both factors: the dynamics could be of additive, or multiplicative kinds, as discussed earlier. One assumption, implicit in investigations of neural network structures, is that some, or all, of the inter-layer interconnection strengths (feed-through and feed-back), are non-null, else the result would be a network structure with islands of (intra-layer) processing with apparently no means of information interchange between layers.

[26] Associative networks, of the automorphic and heteromorphic kinds, have been discussed in the next article.

Neural Network Architectures

Interconnected ensembles of neurons, or neuron-like dynamical systems, while a plausible framework for the study of cognitive capacities exhibited by biological systems, in particular human beings and also for modeling complex system behavior, is too general a framework, even with layering, to be readily useful in neuro-science as well as neuro-engineering. One of the crucial issues in a study of neural networks is to be able to associate classes of neural network structures with appropriate classes of information processing. Architecture,[27] a term incorporated into studies of information processing (computational) systems to mean a description of components involved in a system whose organization is related to the structure, function and performance of the system, the correspondences being conjointly discovered or evaluated, provides a reasonable medium for the study of information processing potential of specific neural network structures.

Neural network architectures, in view of the formal specification introduced in the previous article, essentially describe the processing

[27] Interpretations of the term architecture in computational systems is not without controversy The most common and forceful, of interpretations suggests that architecture is the perception of a system at a microscopic, rather than macroscopic (or molar), level of discourse: specifically in computer systems, architecture is considered to be the assembly (*ie*, hardware) level description of systems as compared to application (*ie*, software) level descriptions (*cf*, **Dasgupta**, 1984). **Baer** (1984), on the other hand, opines that architecture involves an account of organization of components, enunciating the structure, function and performance of the system as a whole This view, while not precluding the former, punctuates the interplay of synthesis and analysis, between the microscopic and macroscopic levels of discourse In the ensuing discussion, I prefer to use the second interpretation.

nodes in number and type and inter-processor interactions in terms of the nature and strength of interconnections (which also specifies layering), and means for deriving the requisite interconnection strengths from the supplied repertoire of examples, *ie*, the *training set*, also known as *knowledge base*, of the required processing to be incorporated (realized) by the network. The dependence of interconnection strengths on training samples, essentially a representational issue, is specified through a *learning procedure*: the learning process is, in general, iterative and it is of importance to guarantee the convergence of the particular procedure used. In the following, I will briefly present some of the prominent architectures from a signal processing perspective, focussing on the functional aspect of information processing provided by a neural substrate. A common trend in presentations of neural network investigations is to refer an appropriately ordered collection of interprocessor (*ie*, inter-neuron) interconnection strengths as the architecture of the network.

Architectural types of neural networks are classified on the basis of the nature of associations, number of layers, recurrent dynamics and adaptivity in interconnection strengths. Networks can be *auto-associative*, or *hetero-associative* depending on whether, or not, the space of patterns in the input is the same as that in the output. In both cases, the association sought could be bi-directional (invertible), though such a requirement is very rare and unless otherwise mentioned, bi-directionality of association will be assumed absent. The number of layers can be either single, or multiple and recurrence in networks, if present, is either of the laterally interacting kind (with one, or more, layers exhibiting intra layer interactions), or due to specific inter-layer feedback connections

I assume that the interconnection strengths of all nodes need to be learnt and hence training has not been invoked as a feature for classification, though adaptivity of the interconnection strengths during usage of the networks has been used as an important criterion for classification Networks wherein interconnection strengths are not adapted have two distinct phases, that of learning, wherein the interconnection strengths are decided, and of usage, wherein the learnt interconnection strengths are employed to provide suitable decision functions. Differences in processing units, if any, is not being explicitly indicated to simplify the understanding of functional characteristics. Note that while neurons have been formulated as dynamical systems, none of the existing (prominent) architectures, make explicit use of neural dynamics and dynamics, if any, in network response is due to inter-node interactions alone.

Perceptrons (*cf*, **Rosenblatt**, 1958), the first (and non-trivial) architecture to be proposed, are hetero-associative, non-recurrent networks wherein the decision units (neurons) are organized in single or multiple layers, the interconnection strengths being non-adaptive. With hardlimiter activation function in all nodes, the network is viewed as representing Boolean functions (formulae of propositional calculus), and also as inducing decision regions in the input space. As discussed earlier, layering is needed to increase the assurance of representation, especially when the decision functions to be represented involve separation by nonlinear manifolds. Multi-layered feed-forward networks[28] have been the subject of prolonged investigation.

Historically the first architectural innovation subsequent to perceptrons, provided by **Kohonen** (1972), initiated lateral interaction into neural information processing. Known in the literature as *Kohonen layer*, the network structure consists of a single layer of (laterally) interacting nodes, with symmetric (*ie*, unprioritized) interprocessor interactions, the profile of interconnections being derived by an 'On Center, Off Surround' rule, compelling near neighbour interactions to be supportive, if not excitatory and interactions of (not too) distant neighbours to be inhibitory, if not ignored. As each processing unit attempts to maximize its own output and at the same time suppress activity in other nodes, this network structure is also characterized as being based on the principle that the *winner takes all*.

It is interesting to note that this hetero-associative network structure, in view of the *competitive* nature of information processing, pro-

[28] These networks are also termed backpropagation networks in the literature, to indicate that learning is accomplished by error back-propagation. However, I will refrain from using this term as it does not explicitly indicate the organizational specific of layering and non-recurrent interconnections. Further, backpropagation, as a learning mechanism, is not restricted to multi-layered feed-forward networks alone.

vides a generalization of the notion of multivibrators (**Millman & Halkias**, 1967), the basis of sequential circuits in digital electronics and has been incorporated into several subsequent architectures. Kohonen layers have been applied in associative memories (**Kohonen**, 1984), topology feature maps (**Kohonen & Mäkisara**, 1986), hamming net and several other clustering situations. One of the attractions of this network structure is because of the unsupervised (actually semi-supervised) learning procedure

Bidirectional Associative Memories (**Kosko**, 1987; 1988), *Cognitron* (**Fukushima**, 1975), *Neo-Cognitron* (**Fukushima & Miyake**, 1982; **Fukushima**, 1987), *Counter Propagation Networks* (**Hecht-Nielsen**, 1987a) and networks of *Adaptive Resonance Theory* (**Carpenter & Grossberg**, 1987a) are some prominent architectures that make use of competitive information processing (*ie*, lateral interaction layers). The *auto-associative* network structure of **Hopfield** (1982) is similar to Kohonen's lateral interaction layer, however, the state encodings, interconnection profile and type of activation function being different, the nature of dynamics is slightly altered.

In both network structures, settling of the network response to a (stable) attractor is very important. Auto-associative networks, in view of automorphic dynamics, have been used in associative memories, solving problems of optimization and search (*cf*, **Baum**, 1986; **Jeffrey & Rosner**, 1986; **Saylor & Stork**, 1986), and have also motivated the entry of physicists into neural information processing. The computation of auto-associative networks is compared to Ising spin systems (*cf*, **van Hemmen**, 1986; **Amit**, 1989), thereby boosting studies in neural network dynamics.

Bi-directional associative memories, an extension of auto-associative networks, realize hetero-associative invertible (bi-directional) maps. At the organizational level, two lateral interaction layers are invited to interact through inter-layer interactions. Counter propagation networks realize hetero-associative invertible (bi-directional) maps with a single lateral interaction layer and provision made for representing dependencies between inputs and outputs. Neo-cognitron is hetero-associative and consists of multiple layers of lateral interaction: the interconnection strengths are not adapted during network use. This architecture is claimed to be capable of providing visual information processing, with (intra-pattern) translational and rotational invariances. It is of interest to note that no architecture has yet been proposed to realize auto-associative maps without recurrence.[29]

Learning and Generalization in Neural Networks

Neural networks, *ie*, ensembles of interconnected processing (decision making) units, invariably exhibit a dependency, in the realization of

[29] **Usui, Nakauchi & Nakano** (1991), however, have suggested a network for realizing invertible maps.

desired processor functionality, on the interconnection strengths between the constituent processing units and the weightages associated with input and output patterns (signals). Connectionist processing, grounded in this dependence, is commonly projected as an incorporation of available (desired) knowledge in the interconnection strengths: this interpretation necessitates that the interconnection strengths be considered as being *separate* from the processing units.

Equally legitimately connectionist processing can be considered as processor realization achieved by networks of parametrıcally selected processing options. The parameterization (of operation, *ie*, decision making) in the participating processing units is incorporated by interconnection strengths associated with corresponding nodes. This alternative treatment to connectionism, though not yet popular, serves to provide a framework for unifying neural networks with networks of automata, typically *schema* of *Turıng Machınes* and *cellular automata*: the latter two are representative of symbolic computation.

A salient aspect of neural network activity, in all treatments of connectionism, is to be able to relate admissible values of interconnection strengths to the (overall) processing functionality realized. Depending on the context, the automated process of explicating the dependency of processing on interconnection strengths is termed *learning* in situations wherein specific aspects of the processing functionality, typically examples (also prototypes) of the input-output association provided, are available and the task is to recreate these very aspects in the network, or studied under the general heading of *pattern formation* in situations wherein certain symmetries regarding the interconnections strengths are known (speculated), generally through constraints imposed by physicalist reductionist renderings to model (theory) building and the desire is to *explore* the kinds and nature, of processors that can be realized with the specific choice of interconnection strengths [30]

Considerations of computability (*cf*, **Hopcroft & Ullman**, 1989), in conjunction with the urge to maintain compatibility with biological metaphors, constrain the problem of 'learning from examples' to one of stating the necessary and sufficient choices to be made in relation to the interconnection strengths in a network given a *finite* number of instances of the desired processor functionality. This repertoire of instances, also prototype input-output associations, is termed *training set*, or *knowledge base* and in this constrained situation, learning seeks to establish relationships between the interconnection strengths of a chosen network and the available (given) training set.

Neural networks, acclaimed model free estimators and applicable in almost all situations of function approximation (processor realization), to maintain the claims of universality, are expected to be equipped with learning procedures that work reliably regardless of whether, or not, the

[30] Exploration of the influence of interprocessor interconnection strengths on pattern formation is commonplace in evolutionary processing like Hopfield networks, genetic algorithms and cellular automata and provides an attractive environment for recasting search problems.

training set is deterministic, or drawn at random (with an appropriate, possibly unknown, distribution) and can to some reasonable degree state a measure of confidence of learning given the statistical nature of the processing instances in the training set.

Finiteness of the number of instances in the training set restricts specification of processing characteristics to a narrow, possibly (in itself) uninteresting, region of the domain on which the neural network is required to be defined and, consequently, mechanisms of extending the desired peculiarities of processing to regions of the input space not covered by the training set are needed: the process of establishing such extension of function evaluation is termed *generalization*. While function evaluation at input space positions other than those incorporated in the training set is guaranteed, axiomatically, by activation functions chosen to be non-constant with no more than a finite number of discontinuities, as *eg*, in sigmoidal (including hard-limiter) and radial basis functions, extension of function evaluation to realize the processing characteristics specified in the training set imposes constraints on the choice of interconnection strengths in the network.

Extension, in the representation, of the desired processing objective to the (entire) input space is, generally, assured by identifying a collection of instances of association between inputs and outputs which are similar, yet distinct, to those in the training set, however, used for the explicit purpose of (cross) validation of the representation suggested for the training set: this collection is termed *test set*. The task of seeking a representation of the specified knowledge (training set) is continued until evaluations of the processor, whose synthesis is guided by representations, in terms of interprocessor interconnection strengths, sought through learning, over input positions described by instances in the test set are satisificable, *ie*, match (the specification in the test set) to within prespecified limits of tolerance.

2.3 Neural Signal Processing: A thematic reconstruction

Information processing, in particular the perceptual categorization of (visual) patterns, has been a constant focus of neural networks since inception through *perceptrons* of **Rosenblatt** (1958). However, signal processing with neural networks, as a specific research activity, originates in the work on ADALINES and MADALINES by **Widrow** (1959), **Widrow & Winter** (1988). Despite the early origin, the pace of research in neural signal processing has accelerated only in the previous decade and of the many influences encouraging signal processing with neural networks, the tutorial paper of **Lippmann** (1987) is noteworthy.

The essential problem of neural signal processing is to realize the desired processor as a networked ensemble of basic decision stages. Processor representation, related to function approximation, sought through a repertoire of input-output correspondences rather than symbolic/functional forms of dependency relationships, is associated with biological metaphors, notably *learning by examples.*

Neural network research, originating in the work of **McCulloch & Pitts** (1943) and widely acclaimed following perceptrons of **Rosenblatt** (1958), has been attributed, in the literature, to have entered a near dormancy following the sharp criticism offered by **Minsky & Papert** (1969). However, neural networks has resurfaced as a major research activity following new directions provided by **Kohonen** (1984), **Hopfield & Tank** (1985), **Denker** (1986), **Grossberg** (1982), **McClelland, Rumelhart**, *et al* (1986a, 1986b).[31]

Though neural networks research has had automated information processing as a consistent central theme, the underlying research interests have not been identical all through. Prior to the reemergence of neural networks, significant emphasis has been laid on realizing (probabilistic) decision functions with multi-layered neural networks: it is in this context that the powerful remark of **Minsky & Papert** (1969) criticizing the absence of a learning Theorem for multi-layered perceptrons is to be appreciated.

Error back-propagation, a mechanism–attributed to **Rumelhart, Hinton & Williams** (1986)–which provides a reasonably general solution to the problem of learning of weights in multi-layered neural

[31] Signal processing with neural networks, though initiated by **Caianiello** (1961), **Fukushima** (1969), **Hopfield & Tank** (1985), **Kohonen** (1980, 1981) and others, has found wide acceptance only since the tutorial paper by **Lippmann** (1987).

networks and the neural information processing schemes suggested by Hopfield and Kohonen, have triggered two distinct trends in neural signal processing. One trend, due to physicists, yet significant in signal processing, has focused on the information processing potential as a function of the structural encoding, *ie*, the physics of neural networks. In this study, neural networks are necessarily of the recurrent kind and the patterns of evolution of the (neural activation) state vector consequent on structural impositions on the inter-processor interactions are in focus.

Neural networks, viewed in this perspective, have been associated with other similar schemes of interconnected ensembles of (local) oscillators, notably Ising spin models of statistical thermodynamics (*cf*, **van Hemmen**, 1986) and Boltzmann machines (**Hinton, Sejnowski & Ackley**, 1984). The dynamics in such networks have been utilized in formulating search problems, in particular those that involve constraints, *eg*, optimal solutions for the 'Travelling Salesman Problem.' Neuro-biologists and neuro-anatomists have benefitted a great deal from these models in trying to identify specific structures of inter-processor interaction (*cf*, **Peretto**, 1992).

The other research trend is to focus on the representation potential of feed-forward neural networks with a single (hidden) layer of processing, the outputs of the processors of this layer are linearly combined to derive the requisite output. In this form, the problem is close at heart to approximation theorists and the major thrust in the study is to overcome the limitations of the (elementary) perceptron like single layer of processing nodes through the use of different kinds of nonlinear association mechanisms.

A popular choice has been to use radial basis functions (**Girosi** & **Poggio**, 1991), essentially nonlinear functions with localized influence, for implementing the activation function. One of the guiding principles in deciding the suitability of activation functions is to ensure good approximation characteristics and a simplification of the problem of learning of the weights associated with the constituent processors of the neural network.

Restriction of decision making to a single layer (of sufficiently many processors) is not without reason. A justification for this choice is provided by a Theorem of Boolean function representation due to Shannon (**Kohavi**, 1978): any Boolean function is representable in an AND–OR–INVERT processing scheme. **Lippmann** (1987) and subsequently others, have identified these three distinct logical operations in neural networks. The input layer (of terminations followed by weighted channels) provides inversions, the single layer of nonlinear processing incorporates logical conjunctions/disjunctions, as the case may be and the final summation level provides the remaining logical function.

Another, more technical, reason underlying the choice of a single layer of nonlinear processing is based on the computational complexity of the problem of loading the training data (*ie*, learning problem). Stephen **Judd** (1990) has pointed out that as the number of nonlinear processing (decision making) layers increases, the complexity of the learning problem increases, whereby it is optimal, in terms of computational resources, to specify a shallow architecture, *ie*, those with fewer number of layers, but sufficiently many processing nodes in the layers.

In this section, I will present a cursory review of the research relevant to neural signal processing This review, inclined towards neuroengineering as compared to the previous section, will be initiated with a historical perspective of neural signal processing. In view of the fact that research in neural networks is being contributed by investigators from several fields, principally physical sciences, mathematical sciences and the engineering community, several conflicting notations exist, disallowing their concomitant usage. Therefore, I have opted to present this review, as in the case of the previous section, with my own notation, a significant proportion of which has already been introduced earlier.

History of Neural Signal Processing

Neural signal processors, in both of the earlier mentioned trends of research, have been considered, in general, as nonlinear processors with interpretations of shift-invariance and incorporating causality when applied to filtering situations. In addition, neural signal processors have occasionally been discussed in processing contexts involving adaptivity and/or stochasticity. Processors realized with neural networks are classified on the basis of the nature of association (as *hetero-associative*, or *auto-associative*), degree of layering (as *single layered*, or *multi layered*, based on the number of 'hidden' layers of decision making) and the incorporation of recurrence and/or competition in processing.[32]

Adaptive linear elements (ADALINES), essentially linear filters subjected to threshold comparison (sign test), are operationally described by (see **Widrow & Lehr**, 1990)

$$\eta(\underline{x}_i, i) = \underline{w}_i\, \underline{x}_i - \theta_i, \qquad (2.18a)$$

$$y(\underline{x}_i, i) = \sigma_h(\eta(x_i, i)), \qquad (2\ 18b)$$

where $i = 0, 1, \ldots$, σ_h is the hard-limiter (also 1-bit quantizer) function. Despite adaptivity of filter weights, the above scheme corresponds to the formal model of neurons wherein shunting and dynamics are suppressed and linear separability is imposed on the corresponding observation (*ie*, input) space.

This scheme, representative of the earliest attempts in neural signal processing, focuses on hetero-associative maps (with no recurrence and competition) and the problem of learning weight values given a training set is looked upon as the operationally equivalent task of adapting

[32] Though dynamical neurons are not unknown, no major processing scheme employing such processors has yet been studied and dynamics in processor response has always been incorporated through recurrence or competition. It is noteworthy that though recurrence and competition share a great deal of commonalty in abstraction, the differences in interpretative content necessitates these two to be viewed as distinct concepts

(filter) weights $\underline{w}$ to a (random) sequence of patterns drawn from the training set. In signal processing applications the adaptation of filter weights, in accordance with the (time) history of function approximation error, is commonly by a least squares approach, typically the 'Least Mean Squares' (LMS) algorithm, often called 'Delta Rule' (*cf*, **Widrow & Hoff**, 1960), though this algorithm is not the most popular in adaptive signal processing as convergence is not guaranteed in any sense stronger than that of expected error.[33]

Since single ADALINES, *ie*, linear classifiers, have a limited representation potential (in terms of separation), as pointed out by **Cover** (1965) and **Minsky & Papert** (1969) and subsequently others, nonlinear approaches have been considered essential for better classifier representation. Of these only two major traditions of realizing hetero-associative (non-recurrent, non-competitive) maps will be considered.

One approach, originally due to **Specht** (1967b) and **Ivankhnenko** (1971), relies on subjecting suitably preprocessed versions of the input signals (patterns) to linear classification: the preprocessing is chosen to impose polynomial transformations on the input pattern vector, thereby presenting second and/or higher order correlations to the linear classifier. Linear classification on polynomially transformed input vectors

[33] Adaptation algorithms for linear (and nonlinear) filters have been discussed extensively in the literature on signal processing. Commonly the 'Recursive Least Squares' (RLS) algorithm (**Haykin**, 1984) is used in parallelized form (as PRLS algorithm due to its superior convergence characteristics. **Chaturvedi** (1994) has shown that PRLS schemes provide a unifying thread for RLS and LMS approaches, and, in this unified framework, has compared the relative performances of the two schemes.

introduces *curvature* in the separation surface, *ie*, allows the decision regions to be non-convex, though simply connected.

Known in the literature as *polynomial neurons* and *higher-order neurons* (**Spirkovska & Reid**, 1992), the operational form of linear classification on polynomially preprocessed inputs is given by

$$\eta(\underline{x}) = \sum_{i=0}^{N} \sum_{j} w_{i_j} P_{i_j}(\underline{x}), \qquad (2.19a)$$

$$y(x) = \sigma(\eta(x)), \qquad (2.19b)$$

where, $P_{i_j}(\underline{x})$ refers to the jth enumeration of homogeneous polynomials of degree i in the elements of $\underline{x}$ (*ie*, *rational varieties* of order i), N refers to the largest degree, possibly infinite, of polynomials relevant to the specific approximation task at hand and σ is the familiar sigmoidal, or hard-limiting nonlinearity. The form for assigning η corresponds, closely, with Volterra filters used in nonlinear signal processing

Discrimination need not be provided by nonlinear activation functions alone. In fact, nonlinearities in the weighting mechanism too can offer interesting discrimination even if the activation function is linear. The following is generalized (steady state) model of isolated neurons.

$$\eta(\underline{x}) = \sum_{p=1}^{P} \prod_{j=1}^{p} (\sum_{i=1}^{n} w_{ji}(x_i)), \qquad (2.20a)$$

$$y(\underline{x}) = \sigma(\eta(\underline{x})), \qquad (2.20b)$$

where, a polynomial discrimination of order P is being assumed, n is the number of channels (inputs) and w with the relevant indices is the functional interconnection strength.

Linear discrimination is obtained when w, in all the channels, is operationally equivalent to multiplication by a (channel specific) constant, and $P = 1$. Though the above expression depicts an instar neuron, a similar generalization to instar-outstar neurons is not difficult to visualize. Functional link nets of **Pao** (1989) are based on instar-outstar neurons with a similar generalization, wherein discrimination is of order 1 and the functional interconnection strengths, generally drawn from the space of trigonometric, or exponential, functions, incorporate *spectral synthesis* as an essential ingredient of neural information processing.

Davidson & Hummer (1993) have pointed out that processors with the functional form

$$y(\underline{x}) \triangleq \eta(\underline{x}) = \bigvee_{i=1}^{n} w_i \wedge x_i, \tag{2.21}$$

where, $\wedge$ and $\vee$, respectively, denote the Minkowskian operations of minimization and maximization (*cf*, **Serra**, 1982), when interconnected in a manner similar to conventional neural networks, are capable of representing morphological operations. Morphology neural networks, as they are termed have been suggested for image processing applications.

The second approach in nonlinear classifier representation is to consider a layering of decision making stages: each layer is composed of adequate number of linear classification elements (neurons). Known as feed-forward networks and MADALINES (for Many ADALINES), such layered ensemble of decision elements[34] induce non-convex partitions and **Lippmann** (1987) points out that larger the number of layers, greater is the possibility of representing fragmented dichotomies, *ıe* dichotomies with disconnected components. It is not difficult to visualize that layered networks, with sufficiently many layers, are capable of representing all functions realized through polynomial neurons.

Widrow, Winter & Baxter (1988) establish that MADALINES with a majority logic at the final stage captures rotational and translational invariances in patterns, however, it is essential that the multitude of hypotheses related to the several translated and rotated versions of the pattern to be recognized be detected in distinct networks In principle, this scheme is no different from that used in array-detectors (*cf*, **Proakis** (1989)), which resemble 'Linear Discrımınant Functions' suggested by **Nilsson** (1965).

Networks of polynomial neurons have also been shown to incorporate translational and rotational invariances (*cf*, **Spirkovska & Reid**, 1992), though, in this case, explicit detection of distınct translated and rotated versions of the patterns, and the subsequent majority logic are

[34] **Aleksander** (1983a) and **Stonham** (1983) discuss pattern dıscrimination and recognition, in the context of patterns described over Boolean (bivalent) spaces, through networks of memory elements: in these networks, the role of neurons (ADALINES) are realized with (programmable) memories storing the essential nature of the ınput-output association. The equivalent of learning is accomplished by identifying the appropriate contents at the various 'addressable' locations of the memory elements

not needed. Invariances, in pattern recognition, are attributed to specific higher order correlations of the input signal (pattern)

Hetero-associative neural signal processors have not been restricted to classifier representation and realization through non-recurrent, non-competitive means. Feed forward neural processors have been discussed in the general context of function approximation, of which classifier representation is a specific case. In this context, the nonlinear activation function σ is generally not discrete valued, but takes on a continuum of values: often a sigmoidal function. Studies on the approximation potential of neural networks and related convergence issues, have brought to light the importance of the nature of nonlinearity in the activation function.[35]

These studies (see **Girosi & Poggio** (1991)) have revealed that nonlinearities with a global influence, sigmoidal function being a typical example, are unsatisfactory for function approximation, with a single layer of decision making, as the number of decision making units (*ie*, neurons) is undesirably large.[36] As a consequence, the convergence characteristics and the assurance of approximation are not adequate.

Present research on neural function approximation focus on networks with a single layer of nonlinear processing and exhibit an emphasis on nonlinear functions with local influence, typically radial ba-

[35] However, such studies are largely limited to single (hidden) layer networks.

[36] This notion has been captured more precisely by **Cybenko** (1989) in terms of the denseness, of the space of functions realized, in the space of continuous functions.

sis functions (*op cit*),[37] to facilitate simpler (compact) representation of functions inducing partitions with non-convex pre-images Gabor functions and Wavelets (**Chui**, 1992; **Daubechies**, 1992), local functions lately popular in signal representation and processing, have also been suggested (see **Daugmann**, 1988 for use of Gabor functions and **Zhang & Benveniste**, 1992; **Pati & Krishnaprasad**, 1993 for the adoption of Wavelets) as suitable candidates for use as activation functions σ of neurons.

Radial basis function networks (*cf*, **Poggio & Girosi**, 1990; **Haykin**, 1994) differ from the neural network architectures discussed earlier in the sense that discrimination is effected non-monotonically on discriminants having quadratic, rather than linear, variation with input patterns. Inspired from a consideration of approximation using regularization theory, these networks approximate the desired function as a member of the linear span of Greens's functions of an appropriately chosen self-adjoint differential operator: the basis functions turn out to be Gaussians when the chosen differential operator is translationally and rotationally invariant. Relative ease in parameter specification and compact network structures have made radial basis function networks popular in signal processing contexts

In view of the fact that the discriminants in radial basis function networks are quadratic functions of the inputs, typically formulated as

[37] Ridge functions of **Ya Lin & Pinkus** (1993) are functionally similar to radial basis functions, with similar approximation characteristics.

the (Euclidean) norm of translated versions of the input pattern, the decision regions of isolated processing nodes are (hyper) spheres in the (Euclidean) space of inputs, the center being specified by the translation vector involved in evaluating the discriminant. Such decision units have been termed as being *diameter limited* by **Minsky & Papert** (1969).

Networks of such processing nodes have been shown (*op cit)* to be incapable of representing predicates of connectedness, relevant in perceptual processing stemming from computational geometry. *Wavelet networks* suggested by **Zhang & Benveniste** (1991, 1992) and **Pati & Krishnaprasad** (1993) too exhibit a diameter limitedness and suffer from the same limitations These limitations while inconsequential in the context of function approximation, are important when the approximation is given a cognitive/perceptual connotation.

Dynamics in the response of neural signal processors, generally incorporated through competition or recurrence, have been initiated by **Kohonen** (1984) and **Hopfield & Tank** (1985), respectively. Of these, competitive networks have been considered mainly in situations demanding interpretation in terms of feature extraction and (unsupervised) clustering, while recurrent networks are employed in realizing automorphic transformations, preferably lacking ergodicity, common in modeling of physical phenomena and in solution of search problems with a requirement of optimization.

These networks are functionally similar to Equation 2.16a, however, differ in the manner of state encoding and types of nonlinearities.[38] Such networks have been studied as *associative memories* wherein pattern recall given (partial) cues has been likened to human memory. Competitive networks provide hetero-associative maps, while recurrent networks, of the Hopfield kind, realize auto-associative maps. Each node in a competitive network is identified with a distinct concept (cluster) and it is not uncommon to find this association interpreted in the same sense as *grand-mother neurons* (*cf*, **Hofstadter**, 1979)

[38] Kohonen networks assume unipolar bivalent neurons, *ie*, output restricted to the limits (also steady state values) '0' and '1', with saturating linear functions for σ, while Hopfield's circuit assumes bipolar neurons (outputs vary between '-1' and '1'), unipolar weights and hardlimiting (or sigmoidal) activation functions The inputs of Kohonen networks are organized on the unit (hyper sphere), whereby partitions are measured in terms of the (solid) angles at the centroid of the (hyper) sphere and the performance of the processor, expressed in terms of the settling (*ie*, convergence) characteristics, is strongly influenced by the (angle of) separation between mutually distinct clusters In contrast, the inputs of Hopfield's circuit are organized on the extended (hyper) cube $[-1, 1]^n$ and as established by **Amit** (1989), settles to an appropriate attractor (fixed point), depending on the initial location of the state vector, only if the transformation incorporated is non-ergodic A weight matrix expressed as the superposition of the self outerproducts of the desired attractors, subject to an annulment of the main diagonal, has been shown to be sufficient to represent (nearly) orthogonal attractors and empirical investigations have revealed that the number of attractors that can be stored with a reasonable degree of recall is of the order of 15% of the number of participating nodes Both networks consider information propagation delays between lateral nodes to be of unit magnitude and impose symmetry in the interaction between past outputs and current evaluation through symmetry in the matrix of weights ϵ and this symmetry together with an identicality of the nonlinear activation function at the various nodes reflects the homogeneous nature of processing In these networks, termed associative and explored with connotations of memory (*ie*, storage and recall), convergence of the dynamics is with respect to an objective function which has been shown, in the case of Hopfield's circuit, to be related to the Hamiltonian (Lyapunov function) familiar in studies of Ising spin systems and this aspect is exploited when these networks, or their variants, are employed in (optimal) search problems, *eg*, solution of 'Traveling Salesman Problem,' or routing in VLSI circuits.

Narendra & Parthasarathy (1990) have discussed dynamical processors obtained by looping back the output of a feed-forward network (of conventional neurons)· this form of computation, a generalization of recurrence considered in Hopfield's circuit, necessitates the number of outputs to be the same as that of inputs. Recurrent computation with probabilistic state transitions have been the focus of *Boltzmann machines*, suggested by **Hinton, Sejnowski & Ackley** (1984) (also see **Hecht-Nielsen**, 1990; **Haykin**, 1994), which is essentially a Hopfield circuit wherein transition of states (*ie*, response of activation functions σ which are allowed to take discrete values -1 or 1) in the processing nodes is governed by the Boltzmann distribution:[39]

$$P(\sigma(\eta_j) = -\sigma(\eta_j)|\eta_j) = \frac{1}{1 + e^{-\Delta H_j/T}} \triangleq \sigma_s(\frac{\Delta H_j}{T}) .$$

A further generalization of recurrent computation has been incorporated in the *Bidirectional associative memories* (BAM) of **Kosko** (1987),

[39] In this expression j and i indicate indices on the single layer of m_1 processing nodes, H, the Hamiltonian, or energy function of the Boltzmann Machine is given by

$$H = -\frac{1}{2}\sum_{j=1}^{m_1}\sum_{\substack{i=1 \\ j\neq i}}^{m_1} w_{ji}\,\sigma(\eta_j)\,\sigma(\eta_i) ,$$

$\Delta H_j = -2\sigma(\eta_j)\eta_j$ is the change in the energy function of node j while flipping the state and T has the connotation of temperature, whose (controlled) reduction freezes the transitions and σ_s is the unipolar sigmoidal function commonly used as the activation function of neural processing elements. This network, trained by a procedure of *simulated annealing* (**Kirkpatrick, Gelatt & Vecchi**, 1983), though slow in convergence, overcomes, in a statistical sense, the annoying aspect of search settling in *local minima* rather than the *global minimum*—common in learning through error backpropagation and in computations of Hopfield circuit. As the convergence of this network is in distribution, these machines have been regarded in the literature to be useful in learning probabilistic maps.

essentially a 2-layer network described, recursively, by

$$\begin{aligned} y^{(2)}_{j^{(2)}}(i) &= \sigma(\underline{w}_{j^{(2)}}\ \underline{y}^{(1)}(i-1) + \theta^{(2)}_{j^{(2)}}),\ j^{(2)} = 1, 2, \quad , m^{(2)}, \\ y^{(1)}_{j^{(1)}}(i) &= \sigma(\underline{\tilde{w}}_{j^{(1)}}\ \underline{y}^{(2)}(i-1) + \theta^{(1)}_{j^{(1)}}),\ j^{(1)} = 1, 2, \ .\ , m^{(1)}, \\ & \qquad i = 0, 1, \ldots, y^{(1)}(i) = y^{(2)}(i) \equiv 0, \forall i \leq 0, \end{aligned}$$

where $[\underline{w}_1, \underline{w}_2, \quad \underline{w}_{m^{(2)}}] \triangleq \boldsymbol{W} = \tilde{\boldsymbol{W}}^{\mathsf{T}} \triangleq [\underline{\tilde{w}}_1, \underline{\tilde{w}}_2, \ldots \underline{\tilde{w}}_{m^{(1)}}]^{\mathsf{T}}$ (noting that the weight matrix $\boldsymbol{W}$ $(\equiv \tilde{\boldsymbol{W}}^{\mathsf{T}}) \in \Re^{m^{(1)} \times m^{(2)}}$), specifying (hetero) associative maps between distinct (or dissimilar) processing fields,[40] and focusing on identification of the possibility of finding convergent patterns in one field, given (partial) cues in the other.

Bidirectional association, in particular the realization of continuous maps, whose inverse map exists and is continuous, have been the focus of *counter propagation networks* (**Hecht-Nielsen**, 1987b; 1990), which incorporates a single decision making layer of the competitive kind and organizes the (bidirectional) association between inputs and outputs through stages of instar and outstar processing (*cf*, **Grossberg**, 1982). the most remarkable aspect of this network is that only one level of conceptual entities encode, simultaneously, a function and its inverse map. A similar attempt at conjoint representation of functions and their inverse maps, though with feed forward networks, has been described by **Usui**, **Nakauchi** & **Nakano** (1991). Both Hecht-Nielsen and Usui *et al* rely on the sufficiency of a single level of decision making (termed

[40] Field theoretic investigations into neural information processing have been discussed by **Amari** (1983)

three layer networks) for representing the desired functions and their inverse maps.

Having seen processors of the hetero-associative and auto-associative kinds through feed-forward, competitive and recurrent schemes, it is natural to get interested, for the sake of completeness, in processors which incorporate interaction between layers, each with lateral dynamical interaction. Such processors have been investigated by **Fukushima** (1975, 1987) through *cognitron* and *neo-cognitron* and **Carpenter & Grossberg** (1986b) in their *Adaptive Resonance Theory* (ART): both efforts have been in visual pattern recognition and induce hetero-associative maps.

Cognitron and Neo-cognitron, essentially a hierarchy (*ie*, feed-through) of lateral interaction layers (generally six in number), are claimed to incorporate translational and rotational invariances. Networks of ART consist two key competitive layers, each influencing the other, unidirectionally through an appropriate feed-forward structure. By design, these networks incorporate storage, search, comparison and recall of patterns and are geared to handle adaptive pattern recognition situations by storing (or replacing appropriate stored patterns with) presented patterns if the mismatch (distance) with any of the patterns already in storage exceeds a threshold: this threshold is ascribed an interpretation of the level of *vigilance*.

Representational Issues in Neural Signal Processing Architectures

The architecture of neural networks used for signal processing resembles, largely, a multi-layered neural network, except for the activation functions of the final layer, which are all identity maps and, hence, trivial from the point of view of categorization: such a formulation is equivalent to stating that the outputs of a neural signal processor are linear combinations of the outputs of a multi-layered neural network (with non-trivial activation functions).[41] Networks with a single layer of processing realizing functions of the form[42,43]

$$f(\underline{x}) = \sum_{i=1}^{m_1} \alpha_i \sigma(\underline{w}_i \cdot \underline{x} - \theta_i)\,, m_1 = 1, 2, \ldots, \tag{2.23}$$

have been of principal focus in studies available in the literature (*cf*, **Hecht-Nielsen**, 1987a; **Cybenko**, 1989; **Mhaskar**, 1993; **Ya Lin & Pinkus**, 1993).

[41] It is common for the architecture of a neural network to be identified by a description involving an ordered list of the kind m_0–m_1– $\cdot m_L$, where L refers to the number of layers of decision making, m_0 the number of elements in patterns incident on the input layer of the network and m_i, $i = 1, 2, \quad , L$ indicate the number of processing nodes in the ith layer of decision making.

[42] While a scalar function has been indicated, this form can be effortlessly extended to cases wherein vector outputs are needed.

[43] In the original formulation, the outputs of dynamical processors like Hopfield's circuit, BAM and Boltzmann machine, in contrast with counter propagation networks, are not expressed as linear combinations of responses of decision elements. A reformulation of these processing structures into the framework suggested by the associated equation is not only obvious, but also offers an opportunity for a (minor) generalization of the original formulations

Networks of the functional form in Equation 2.23nopage, uneasily identified with three layer neural networks, are claimed, in the literature, as being sufficient to represent all functions of interest: this claim is supported by the rigorous exposition of **Cybenko** (1989), followed by **Vepsáláinen** (1991), **Mhaskar** (1993) and others Functions realized (as in the above networks) through finite, though unrestricted, linear combinations of the outputs of a single (hidden) layer of processing, have been shown through Cybenko's efforts as being dense in the space of continuous functions.

The activation functions σ, in Cybenko's claim of denseness of representation, are chosen to be sigmoidal, thereby asserting the existence of representation, with *arbitrary* accuracy, for all continuous functions; however, the network structure might involve an unappealingly large number of nodes in the (single) processing layer. The universality of neural networks, though with a single layer of processing, in approximation has been established by **Hornik, Stinchcombe & White** (1989), wherein continuity, together with non-constancy, of the activation functions has been shown to assure denseness of representation in the space of continuous functions.

Hecht-Nielsen (1987c) and later **Shrier, Barron & Gilstrap** (1987), **Girosi & Poggio** (1991), **Cotter & Guillerm** (1992), **Kůrková** (1992) and others have studied the representational potential of networks involving two layers of processing, again identified with three layer neural networks, realizing functions of the form

$$y(\underline{x}) = \sum_{j=1}^{m_2} \beta_j \sigma_j^{(2)} (\sum_{i=1}^{m_1} \alpha_i \sigma_i^{(1)} (\underline{w}_{ij}\ \underline{x} - \theta_{ij}^{(1)}) - \theta_j^{(2)}) \tag{2.24}$$

Of particular interest in these studies is a function representation Theorem due to **Sprecher** (1965), an improvisation on that established by **Kolmogorov** (1957b) in connection with a solution (in the sense of a denial of the hypothesis) of the 13th problem of Hilbert

The Theorems of Kolmogorov and Sprecher suggest an approximation scheme similar, in form, to the above equation and, on the strength of this similarity of form, a representation of all continuous functions described on a bounded linear subspace (of $n(\equiv m_0)$ dimensions) of the Cartesian product space $\Re^n$, a typical choice being $\mathcal{E}^n$ the Euclidean space $[0,1]^n$, is claimed to be admitted by a network comprising exactly $n(n+1)$ nodes in the first layer of processing and $\overline{2n+1}$ nodes in the second (and final) decision making layer

Issues in Concept Representation

Representation of perceptually relevant operations is the principal focus of neural networks and it is common to find functions represented by nodes, in the ensemble being associated with concepts. In this spirit, a layered (feed-forward, non-evolutionary) neural network is interpreted as realizing a hierarchy of concepts. Typically, concepts are expected to highlight specific relative organization of assignments in the incident input patterns and concepts are often interpreted as predicates of logic, generally of zeroth[44] order (*ie*, propositional calculus), operating on elements within the input pattern.

Neurons with hard-limiting activation functions, studied originally by **McCulloch & Pitts** (1943), and their networks have been shown to represent Boolean functions, essentially formulae of the propositional calculus. Sigmoidal activation functions have been shown, in the literature, to enable a representation of formulae in the calculus of *fuzzy* propositions noting that the graded response provided by such activation functions is an excellent candidate for being a (set) membership function.

At a conceptual level, discrimination due to piece-wise monotonic activation functions, which are described, essentially, on localized (possibly compact) support, is similar to that provided by sigmoidal activation functions, in that, the response of (isolated) neurons is a statement of the occurrence of specific relative organization of assignments in the relevant input patterns. Hence neurons with such activation functions are considered to represent predicates of an appropriate logical system.[45]

[44] Predicates of zeroth order are more technically known as propositions. I have preferred to use the more general term predicates to maintain reasonable compatibility with the usage of the term predicates by **Minsky & Papert** (1969). At an abstract level, predicates really are relations defined to capture specific logical correspondences between relevant entities: in the present discussion, logical association is sought between elements of (input) patterns.

[45] Predicates represented by piece-wise monotonic activation functions generally rest on supports that have several, mutually non-overlapping, connected regions within the

Specificity of relative organization is decided completely by the parameters (mainly weights, thresholds – as incorporated by the abstract translation function b – and, relatively infrequently, the abstract amplification function a) and thus, an interpretation of parameters as *templates* of the perceptual entities being sought to be represented would not be inappropriate. Neural signal processing then is akin to template matching, however, the crux of the problem lies in deciding, by automatic means, relevant templates given (valid) examples of association between instances of input patterns and (perceptually grounded) responses, actions, or decisions.

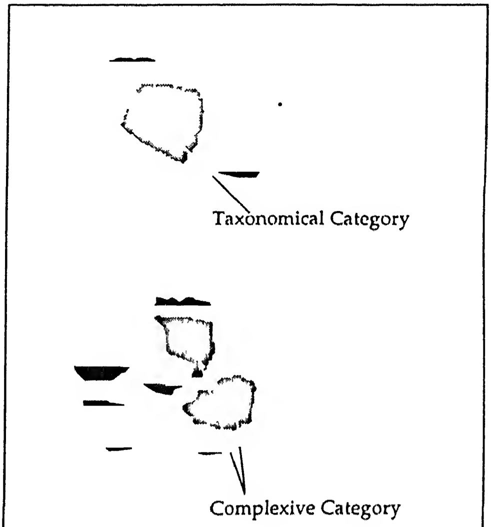


Figure 2.4· Types of concepts

Concepts, identified essentially as categories, are distinguished by **van Loocke** (1994) as being *taxonomıc* or *complexive* This distinction is made on the basis of the existence of a common core of attributes, or features, in the instances, or examples, meant to suggest the concept. Implicit is the assumption that no formal description of the concepts, or equivalently cat-

input pattern and distinct (connected) components of the support correspond to distinct monotonıc segments of the activation function egories, is available and the only description that can be given is in terms of (illustrative) examples. In this framework, a category having instances containing a (non-trivial) common core, or essence, is considered as taxonomical and the concept, or category label, is associated with this common core, whereas, complexive concepts refer to the absence of commonalty in the instances and the labels of such concepts have to necessarily be linked to all instances

A majority of neural network architectures and situations of engineering interest, are concerned with the representation of taxonomical concepts, and the sole effort of generalization, during training, is to derive the (relevant) common core given the examples of input-output association, *ie* knowledge about processor functionality. Complexive concepts arising in information processing contexts like natural language understanding, or pre-attentive vision, essentially situations wherein the (representative) instances have semantic import, rather than syntactic relevance (as in taxonomical categories) and their representation has been focused in the architectures of ART.

2.4 Summary

Representation of concepts, mainly of the taxonomical kind, is the focus of information processing and the continued increase in the involvement of automated information processing and, ultimately, the automation of intelligence, in nearly every aspect of human existence, has necessitated information processing, typically decision making and estimation, to be supported in situations wherein the conditions ensuring satisfactory performance of linear procedures cannot be guaranteed and exhaustive formal (symbolic) statements of processing functionality are almost impossible to be enunciated. Nonlinear processing approaches have been sought to overcome the serious limitations of linear approaches and of these neural network based schemes, originally investigated as models of (human) abilities, are prominent.

Connectionist information processing systems, essentially neural network schemes of processor realization designed to represent the inherent structure specified through examples of input-output associations rather than merely recreate mappings recorded in a training set (as is the case in the approach based on the formalism of Turing Machines), provide a natural framework for the synthesis of information handling when perceptual and/or cognitive interpretations are attached to the processing steps. This framework is sufficiently abstract and universality of processor representation, generally considered the sole preserve of Turing Machines – and other automata equated to this formalism by Church's thesis (*cf*, **Lewis & Papadimitriou**, 1981) – cannot easily be denied to neural information processing. (However, adequate comparative evaluation of neural networks in relation to Turing Machines, in the sense of equivalence, is not yet available.)

Chapter 3

Processor Representation in Isolated Neurons

A representation . implies the existence of two related but functionally separate worlds: the *represented world* and the *representing world* .... In order to specify a representation completely, . one must state: (1) what the represented world is; (2) what the representing world is; (3) what aspects of the represented world are being modeled; (4) what aspects of the representing world are doing the modeling; and (5) what are the correspondences between the two worlds. A representation is really a *representational system* that includes all five aspects.

—Stephen E Palmer
in *Fundamental Aspects of Cognitive Representation*,
Chapter 9 of *Cognition and Categorization*,
edited by Eleanor Rosch and Barbara Lloyd,
Lawrence Erlbaum Associates, Publishers,
Hillsdale, New Jersey, 1978

A study of representation of (signal) processors in isolated neurons is essentially a study of mapping the input and output spaces of the processor into those of the isolated neuron and choosing appropriate weight and threshold values so as to capture the required association between the input and output spaces in terms of the association mechanism characteristic of the neuron. The dependency of outputs on inputs in isolated neurons is commonly expressed as a nonlinear evaluation of a projection of the input: the weight influences the projection and the threshold participates in the nonlinear evaluation

Projections induce a partitioning on the domain and consequently processors incorporating projections in their functionality establish mappings that associate multiple inputs to the same output value. In such a processing situation relative evaluations between outputs (decisions) don't, in general, reflect a relative assessment of corresponding inputs. However, by a suitable restriction of the input space – the nature of restriction is not independent of the class of projection operators – the desired preservation of relationships between inputs in the corresponding outputs is achieved.

In the following I establish the existence of weights that preserve discrete spaces in a one-dimensional (linear) subspace of $\Re^n$. These weights by mapping functions defined on discrete spaces to sequences reduce the problem of learning to one of an enumeration of the weight and a problem of search, in a linear order, for the threshold. I also establish that the notion of preservation is independent of the radix of numbering and identify, through constructive procedures, with every non-null weight in $\Re^n$ the existence of a discrete subset of $\Re^n$ which is preserved in a one-dimensional linear space

The approach in this thesis is to provide an analysis that is independent of the interpretative framework and all signals (patterns) will thus be considered as being no more than members of vector spaces of appropriate dimension Input space preservation, established initially on binary vector collections and subsequently extended to more general discrete spaces, allows the identification of a subspace that will allow a parameterized description of the realized functions. Such an identification facilitates an easy characterization of the representation of processors in isolated neurons and networks of neurons.

Input space preservation and identification of certain discrete spaces preserved in one-dimensional spaces are considered in § 3.1: this discussion, though initiated in the context of isolated neurons, is relevant for networks of neurons too.[1] The implications of preservation on function representation, in particular, linear separable dichotomies, is studied in § 3.2 (*p*. 138). Learning and generalization issues, as influenced by preservation of input spaces, are studied in § 3.3 (*p*. 154). Extension of the notion of preservation to more general discrete spaces, is taken up in § 3.4 (*p*. 171).

[1] Representational issues in layered networks of neurons are considered in Chapters 4 and 5.

3.1 Preservation of Discrete Input Spaces

Consider the formal model of an isolated neuron in steady state:

$$\eta(\underline{x}) = \underline{w}\,\underline{x} - \theta, \qquad (3\ 1a)$$

$$y(\underline{x}) = \sigma(\eta(\underline{x})) \qquad (3.1b)$$

In the above model, the input vectors $\underline{x}$ are considered presented to the neuron from a subset of $\Re^n$, $n = 1, 2,\quad$, neuron weights $\underline{w}$ are drawn from $\Re^n$ and the nature of outputs $y \in \mathcal{Y}$ is decided by the choice of σ as described in Chapter 2. I will begin by considering the case wherein inputs are presented from $\mathcal{B}^n \triangleq \{-1, +1\}^n$, clearly a discrete subset of $\Re^n$ which has the vector $\underline{0} \in \Re^n$ as its centroid (or origin).[2]

For notational convenience the space $\mathcal{B}^1 \equiv \{-1, +1\}$ is denoted by $\mathcal{B}$. In the literature related to neural networks $\mathcal{B}^n$, the collection of n-dimensional binary vectors in $\Re^n$, is commonly interpreted as the Boolean hyper-cube of dimensionality n and this term will be used in the subsequent discussion.[3] Elements in the vectors belonging to $\mathcal{B}^n$ are related to statements asserting the presence (or absence) of certain

[2] All elements of the input vector $\underline{x}$ are considered to be compatible and thus only regular structures will be investigated While it is not impossible to associate different symbol spaces with the different elements of $\underline{x}$, such an association would immediately violate the symmetry of reasoning and, hence, this asymmetric choice (which would have, inevitably, led to irregular discrete symbol spaces) has not been made. Thus, the discrete spaces in this thesis will all be based on n-dimensional hyper-cubes, for an appropriate value of n rather than the more general situation of parallelepipeds in n dimensions. Generality in the characterization, however, is not lost as a result of restricting the discussion to regular structures.

[3] A Boolean hyper-cube is the ground set of a Boolean algebra Topologically this set incorporates the geometrical features of cubes in three dimensional Euclidean space

features of interest in the members of the observation space. The features are, however, given by the specific framework of interpretation associated with the input and observation spaces.

Denote by $\mathcal{L}_{\underline{w}}$ the one-dimensional linear subspace[4] (of $\Re^n$) described by a weight $\underline{w}$·

$$\mathcal{L}_{\underline{w}} \triangleq \{\beta \underline{w} \mid \beta \in \Re\}$$

Note that $\mathcal{L}_{\underline{w}}$ is isomorphic to $\Re$, the real line. In Equation 3.1a η involves an evaluation of the scalar product between $\underline{w}$ and $\underline{x}$. As the scalar product induces a partitioning of the input space $\Re^n$ in terms of hyper planes[5] and the role of η is to provide an ordering of these hyper planes through the (natural) order in $\mathcal{L}_{\underline{w}}$, for a choice of weight $\underline{w}$ in the neuron, the following is introduced.

Definition 3.1.1 *A many-to-one transformation, say f· $\mathcal{A}_D \rightarrow \mathcal{A}_R$, is said to preserve all points of a subset A of $\mathcal{A}_D$, $A \subseteq \mathcal{A}_D$, in $\mathcal{A}_R$ if there exists a subset, say A_s, of $\mathcal{A}_R$, $A_s \subseteq \mathcal{A}_R$, with the following properties.*

1. Uniqueness preservation. *All points of A_s can be put in one-one correspondence with A.*

[4] A subspace is a subset of a vector space (commonly $\Re^n$) that is closed with respect to the operations of addition and multiplication by a scalar (*cf*, **Itô**, 1987).

[5] The image (under a translation) of a subspace of a vector space (commonly $\Re^n$ with a one-dimensional quotient space is termed a hyper plane A subset $\pi \subset \mathcal{X}$ is a hyper plane in a vector space $\mathcal{X}$ over a field $\mathcal{K}$ if and only if $\pi = \{x | f(x) = \alpha\}$ for $\alpha \in \mathcal{K}$ and a certain non-zero linear functional $f \in \mathcal{X}$ (*cf*, **Itô**, 1987)

2. Order preservation. *For any partial ordering relation, say $\preceq$, defined on $\mathcal{A}_D$ there exists a partial ordering relation on $\mathcal{A}_R$, denoted by $\tilde{\preceq}$, such that*

$$\forall a_1, a_2 \in \mathcal{A} \qquad (a_1, a_2) \in \preceq \; \equiv (\tilde{a}_1, \tilde{a}_2) \in \tilde{\preceq},$$

where, $\tilde{a}_1$ and $\tilde{a}_2$ are points in $\mathcal{A}_r$ corresponding to a_1 and a_2, respectively.

3. Regularity condition *The set $\mathcal{A}_s \subset \mathcal{A}_R$ is in one-one correspondence with a set of rationals, say $\mathcal{A}_r$, given by*

$$\mathcal{A}_r = \bigcup_{i=1}^{k} \left\{\alpha_i j_i, \alpha_i \overline{j_i + 2}, \ldots \alpha_i \overline{j_i + 2k_i}\right\},\; j_i = \frac{3 + (-1)^i}{2},$$

for some $\alpha_i, k_i = 1, 2, \quad .; i = 1, 2, . \; . k; k = 1, 2, . \; . .$

The sets $\mathcal{A}_D$ and $\mathcal{A}_R$ are not, in general, the same and, hence the ordering relations related to these sets have been considered different. Further, $\mathcal{A}_D \subseteq \Re^n$ and $\mathcal{A}_R \subseteq \mathcal{L}_{\underline{w}}$ from Equation 3.1. While in the latter set 'less than or equal to' (denoted by $\leq$) is a natural ordering relation, no such natural ordering relation exists for $\mathcal{A}_D$ when the dimensionality n is larger than unity. Thus the relation $\preceq$ is assumed given. A preservation of the partial ordering on $\mathcal{A}_D$ in terms of a partial ordering on $\mathcal{A}_R$ is essential to allow relationships between inputs to be preserved in relative evaluations of outputs.

Regularity in the points of $\mathcal{A}_s$ structures the input space $\mathcal{A}$ to be composed of certain unions of spaces whose image in $\mathcal{A}_R$ is a collection of uniformly spaced points. Such a restriction of the input space points provides the representational advantage of specifying the set $\mathcal{A}$ recursively from a basic set that is preserved in an appropriate subset of $\mathcal{A}_R$. Regularity operating in conjunction with order preservation enforces symmetry in the subset $\mathcal{A}$ that is put in one-one corresponaence with $\mathcal{A}_s$, however, this aspect will not be invoked in the present investigation in view of the nonlinear nature of the function operating on $\mathcal{A}_s$.

As the operation of scalar product (also termed inner product) is uniquely identified with weights, input space partitioning induced by inner product will be attributed to the weights used in the operation. The bilinearity of scalar products (*ie*, linearity with respect to inputs given a weight as well as linearity with respect to weights given an input) assures one-one correspondence, order preservation and regularity in certain discrete subsets of the n-dimensional Euclidean space. In the following the existence of operations enabling a preservation of discrete spaces will be considered.

THEOREM **3.1.1** *There exist weights $\underline{w}$ in isolated neurons which preserve, distinctly, all points of the n dimensional Boolean hyper-cube $\mathcal{B}^n$ in the one-dimensional space $\mathcal{L}_{\underline{w}}$ for all n, $n = 1, 2, \ldots$.*

An illustration of the discrete space $\mathcal{B}^2$ and few of the weights that allow a preservation of all points belonging to $\mathcal{B}^2$ in a one-dimensional space in the direction of the weight vector are illustraced in Figure 3.1. This illustration explicitly indicates the aspects of one-one correspondence, regularity and the natural ordering in the points along the direction of the chosen weight.

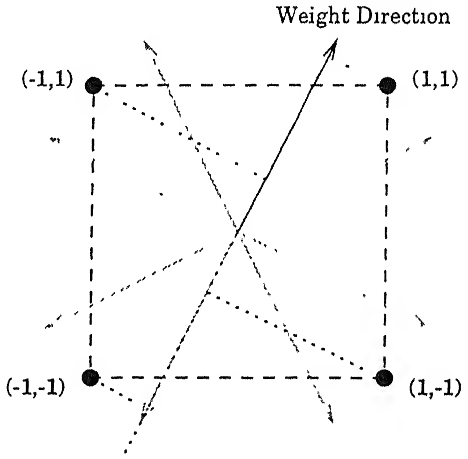


Figure 3.1: Illustration of $\mathcal{B}^2$

Proof: As each of the elements of the input vector $\underline{x}$ takes on one of two possibilities, $+1$ and -1, a weight vector in which the n elements are assigned unique powers of 2 will ensure that the inner product $\underline{w}{\cdot}\underline{x}$ maps the 2^n distinct points of $\mathcal{B}^n$ to distinct points (numbers) in $\Re$. (The weight directions shown in Figure 3.1 conform to this assignment.) For example, a weight chosen as $w_i = 2^{i-1}$, $i = 1, 2, \ldots n$, is a good candidate for establishing a one-one correspondence between $\mathcal{B}^n$ and a discrete subset (of 2^n points) in $\mathcal{L}_{\underline{w}}$.

It is immediately apparent that when $\mathcal{B}^n$, for all n, is interpreted as a poset with the help of a partial ordering relation, say $\preceq$, this partial ordering is preserved in the corresponding points in $\mathcal{L}_{\underline{w}}$, *ie*,

$$\forall \underline{x}_1, \underline{x}_2 \in \mathcal{B}^n \;\; (\underline{x}_1, \underline{x}_2) \in \; \preceq \; \equiv (\underline{w}{\cdot}\underline{x}_1, \underline{w}{\cdot}\underline{x}_2) \in \; \leq \text{ for all } \underline{w} \in \Re^n,\ \underline{w} \neq \underline{0},$$

where, $\leq$ is the relation 'less than or equal to,' also known as 'not greater than ' This preservation of partial ordering is a consequence of the axioms of inner product operation.

Weights given by the assignment $w_i = \pm 2^{i-1}$, $i = 1, 2, .\ \ n$, satisfy the regularity condition required for preservance.

□

The above theorem suggests multiple weights that accommodate a preservation of the same discrete space. In order to facilitate a characterization of these weights, denote by $\mathcal{B}^n(\zeta, \underline{\vartheta})$ the discrete space obtained by scaling and translating the Boolean hyper-cube $\mathcal{B}^n$:

$$\mathcal{B}^n(\zeta, \underline{\vartheta}) = \zeta \mathcal{B}^n + \underline{\vartheta} \ \triangleq \ \{-\zeta + \vartheta_1, +\zeta + \vartheta_1\} \times \{-\zeta + \vartheta_2, +\zeta + \vartheta_2\} \\ \times \cdot \ \ \{-\zeta + \vartheta_n, +\zeta + \vartheta_n\},$$

where, $\zeta \in \Re_+$ is the scale factor and $\underline{\vartheta} \triangleq [\vartheta_1, \vartheta_2, \ldots \vartheta_n]^\top \in \Re^n$ is the translation applied to all points of $\mathcal{B}^n$. (Note that $\mathcal{B}^n \equiv \mathcal{B}^n(1, \underline{0})$ and $\mathcal{B} = \mathcal{B}^1(1, 0)$.) Weights accommodating a preservation of points belonging to $\mathcal{B}^n$ are described as in the following.

THEOREM **3.1.2** *Weights $\underline{w}$ that preserve $\mathcal{B}^n$ in $\mathcal{L}_{\underline{w}}$ are given by the assignment*

$$w_i \in \bigcup_{j=1}^{n} \mathcal{B}(\alpha 2^{j-1}, 0), \ \alpha \in \Re_+, \ i = 1, 2, .\ \ n, \tag{3.2}$$

subject to the restriction that $|w_i| \neq |w_k|$ *for all* $i, k = 1, 2, \ \ldots n$, $i \neq k$.

$\mathcal{P}$ROOF: The property that every non-null weight vector preserves partial ordering under inner products, invoked in the proof of the earlier statement, necessitates only a one-one correspondence between $\mathcal{B}^n$ and a discrete subset of $\mathcal{L}_{\underline{w}}$ to be established. Since weight assignment is in accordance with the prescription given in the earlier statement, the one-one correspondence too is readily evident

Regularity in the collection of points in $\mathcal{L}_{\underline{w}}$ which is put in one-one correspondence with $\mathcal{B}^n$ is not affected by scaling and translation as both operations apply uniformly to all points in $\mathcal{B}^n$.

□

Theorem 3 1.2 suggests the space of possible assignments up to a common positive scale factor: this scale factor has been denoted by α As indicated in Figure 3.1 (*p*. 114) preservance is affected by the direction of the chosen weight $\underline{w}$, the common scale factor – this influences the norm – serving to control the separation between adjacent points corresponding to the image of $\mathcal{B}^n$ in $\mathcal{L}_{\underline{w}}$. Preservance of points belonging to the discrete space $\mathcal{B}^n$ in $\mathcal{L}_{\underline{w}}$ for a weight $\underline{w}$ shows directional dependence as the uniqueness in the images of points in $\mathcal{B}^n$ under inner product is directionally dependent

It is easy to accept preservance of $\mathcal{B}^n$ in $\mathcal{L}_{\underline{w}}$ when the elements of $\underline{w}$ are given by $w_\imath = 2^{\imath-1}$, $\imath = 1, 2, ..\ n$. The following examples illustrate the nature of preservation of $\mathcal{B}^n$ for other choices of $\underline{w}$ governed by the assignment in Theorem 3.1.2. In these examples the common scale factor α has been retained as a variable to highlight the dependence of preservance only on the direction of the weight vector and not the scale factor. For convenience, the examples are restricted to binary collections in Euclidean spaces whose dimensionality is small.

Example 1: *Case* $n = 3$, $w_1 = \alpha$, $w_2 = -2\alpha$, $w_3 = 4\alpha$.

$\underline{x}$	$-1-1-1$	$-1-1+1$	$-1+1-1$	$-1+1+1$
$\underline{w}\ \underline{x}$	-3α	-1α	-7α	-5α
$\underline{x}$	$+1-1-1$	$+1-1+1$	$+1+1-1$	$+1+1+1$
$\underline{w}\ \underline{x}$	$+5\alpha$	$+7\alpha$	$+1\alpha$	$+3\alpha$

Example 2: *Case* $n = 3$, $w_1 = -4\alpha$, $w_2 = 1\alpha$, $w_3 = -2\alpha$.

$\underline{x}$	$-1-1-1$	$-1-1+1$	$-1+1-1$	$-1+1+1$
$\underline{w}\ \underline{x}$	$+5\alpha$	$+1\alpha$	$+7\alpha$	$+3\alpha$
$\underline{x}$	$+1-1-1$	$+1-1+1$	$+1+1-1$	$+1+1+1$
$\underline{w}\cdot\underline{x}$	-3α	-7α	-1α	-5α

Example 3: *Case* $n = 4$, $w_1 = -8\alpha$, $w_2 = 1\alpha$, $w_3 = 2\alpha$, $w_4 = 4\alpha$.

$\underline{x}$	$-1-1-1-1$	$-1-1-1+1$	$-1-1+1-1$	$-1-1+1+1$
$\underline{w}\ \underline{x}$	$+01\alpha$	$+09\alpha$	$+05\alpha$	$+13\alpha$
$\underline{x}$	$-1+1-1-1$	$-1+1-1+1$	$-1+1+1-1$	$-1+1+1+1$
$\underline{w}\ \underline{x}$	$+03\alpha$	$+11\alpha$	$+07\alpha$	$+15\alpha$
$\underline{x}$	$+1-1-1-1$	$+1-1-1+1$	$+1-1+1-1$	$+1-1+1+1$
$\underline{w}\cdot\underline{x}$	-15α	-07α	-11α	-03α
$\underline{x}$	$+1+1-1-1$	$+1+1-1+1$	$+1+1+1-1$	$+1+1+1+1$
$\underline{w}\cdot\underline{x}$	-13α	-05α	-09α	-01α

Without the necessity of a proof, the above examples allow the following to be easily appreciated

PROPOSITION **3.1.1** *Given a weight $\underline{w}$ governed by the assignment in* Theorem 3.1.2 *(with $\alpha \in \Re_+$) such that the Boolean hyper-cube $\mathcal{B}^n$ is preserved in $\mathcal{L}_{\underline{w}}$, the points in the linear sub-space $\mathcal{L}_{\underline{w}}$ which are in one-one correspondence with the 2^n vertices of $\mathcal{B}^n$ are*

1. *identified on the basis of the numerical value of the binary representation as decided by the choice of weights,*
2. *equidistant from adjacent members (implication of regularity) and*
3. *restricted to the interval $\left[-\alpha\overline{2^n-1},\ +\alpha\overline{2^n-1}\right] \subset \Re$*

The positive scale factor α serves to enrich the space of weights that accommodate a preservation of $\mathcal{B}^n$. In the following I will denote the collection of weights $\underline{w}$ that preserve $\mathcal{B}^n$ in $\mathcal{L}_{\underline{w}}$ by $\wp_n$, the suffix n indicating the dimensionality of the input space over which the weights are applicable. For convenience of analysis $\wp_n(\alpha)$ will be used to denote the restriction[6] of weights given α. The set $\wp_n(\alpha)$, for any $\alpha \in \Re_+$ contains weights of identical norm, the norm being a function of α. Weights $\underline{w}$ that preserve all points of $\mathcal{B}^n$ in $\mathcal{L}_{\underline{w}}$ are specified up to a common scale factor by the following.

[6] It is not incorrect to state that $\wp_n(\alpha)$ is the coset $\alpha\wp_n(1)$ of $\wp_n(1)$ in $\wp$.

Corollary to Theorem 3.1.2 *The entire collection of weights $\underline{w}$ that preserve*[7] *$\mathcal{B}^n$ in $\mathcal{L}_{\underline{w}}$ given any $\alpha \in \Re_+$ is expressed as in the following.*

$$\begin{aligned}\wp_n(\alpha) \;&=\; \left(\bigcup_{i=1}^{n} \mathcal{B}(\alpha 2^{i-1}, 0)\right)^{n} \setminus \\ &\qquad \bigcup_{i=1}^{n}\bigcup_{j=1}^{n}\bigcup_{k=1}^{n} \left(\mathcal{B}^{i}(\alpha 2^{j-1}, \underline{0}_i) \times \mathcal{B}^{n-i}(\alpha 2^{k-1}, \underline{0}_{\overline{n-i}})\right),\end{aligned}$$

where, $\underline{0}_j$ is the zero (origin) of $\Re^j$, $j = 1, 2, \ldots,$ and $\mathcal{A}^0$, the 0-fold Cartesian product of any space $\mathcal{A}$, is $\emptyset$, the empty set.

Note that the set

$$\bigcup_{i=1}^{n}\bigcup_{j=1}^{n}\bigcup_{k=1}^{n} \left(\mathcal{B}^{i}(\alpha 2^{j-1}, \underline{0}_i) \times \mathcal{B}^{n-i}(\alpha 2^{k-1}, \underline{0}_{\overline{n-i}})\right)$$

describes weights which have at least two elements with identical magnitude and these are excluded from

$$\left(\bigcup_{i=1}^{n} \mathcal{B}(\alpha 2^{i-1}, 0)\right)^{n},$$

the space of possible weights. ($\mathcal{B}(\zeta, \vartheta) \equiv \mathcal{B}^1(\zeta, \vartheta)\ \forall \zeta \in \Re_+,\ \vartheta \in \Re$.) In the above statement the space of possible weights and the collection of weights that have two or more elements with identical magnitude are expressed in terms of scaled Boolean hyper-cubes.

Proposition 3.1.2 *The number of preservance weights for an input space of n-dimensions, $n = 1, 2, \ldots,$ is given for any $\alpha \in \Re_+$ by*

$$|\wp_n(\alpha)| = n!\,2^n$$

[7] Note that as the weight $\underline{w}$ changes, the one-dimensional space $\mathcal{L}_{\underline{w}}$, by virtue of being the linear sub-space of $\Re^n$ in the direction of $\underline{w}$, also changes correspondingly

The above two statements are immediate consequences of Theorem 3.1.2 (p. 115) and, hence, proofs are not required. Proposition 3.1.2 suggests the number of distinct directions along which weight vectors can be chosen to accommodate a preservation of points belonging to $\mathcal{B}^n$. Noting that weights admitting preservance of a Boolean hyper-cube belong to scaled Boolean hyper-cubes, it is of interest to know the possibility of preserving the scaled and translated Boolean hyper-cubes $\mathcal{B}^n(\zeta, \underline{\vartheta})$ for appropriate values of ζ and ϑ. From the definition of Boolean hyper-cubes, the following is evident.

PROPOSITION **3.1.3** *Weights $\underline{w}$ that preserve the n-dimensional Boolean hyper-cube $\mathcal{B}^n$ in $\mathcal{L}_{\underline{w}}$ also preserve $\mathcal{B}^n(\zeta, \underline{\vartheta})$ in $\mathcal{L}_{\underline{w}}$, $\zeta \in \Re_{+}$, $\underline{\vartheta} \in \Re^n$.*

PROOF: It is simple to observe that the scale factor ζ does not alter the preservation property as long as ζ is non-null. Similarly, a translation of the origin by $\underline{\vartheta}$ adds the component $\underline{w}\ \underline{\vartheta}$ to the inner product and as this addition applies, uniformly, to all nodes of $\mathcal{B}^n$, preservation effected by a weight $\underline{w}$ is unaltered due to scaling and (origin) translation of the input space $\mathcal{B}^n$.

□

Though regularity is unaffected by scaling, the separation between adjacent points forming images of points belonging to $\mathcal{B}^n(\zeta, \underline{\vartheta})$ increases as the positive scale factor ζ. In contrast, translation does not alter the images relative to each other. Note that as a consequence of scaling and translation, the points in $\mathcal{L}_{\underline{w}}$ which preserve the points of $\mathcal{B}^n(\zeta, \underline{\vartheta})$, under a weight $\underline{w}$ governed by the assignment in Theorem 3.1.2 (p. 115), are limited to the interval

$$[-\alpha\zeta 2^{n-1} + \underline{w}\ \underline{\vartheta},\ +\alpha\zeta 2^{n-1} + \underline{w}\ \underline{\vartheta}] \equiv \alpha\zeta 2^{n-1}\,[-1,\ +1] + \underline{w}\ \underline{\vartheta}.$$

These bounds reflect scaling in the weights as well as inputs.

In the following, the points in $\mathcal{L}_{\underline{w}}$ that are put in one-one correspondence with those of $\mathcal{B}^n(\zeta, \underline{\vartheta})$ given $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$ will be denoted by $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{B}^n(\zeta, \underline{\vartheta}))$ where $\alpha \in \Re_+$ is the scale factor associated with the weights involved in the preservation of $\mathcal{B}^n(\zeta, \underline{\vartheta})$ in $\mathcal{L}_{\underline{w}}$. (Note that $|\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{B}^n(\zeta, \underline{\vartheta}))| = |\mathcal{B}^n(\zeta, \underline{\vartheta})| \triangleq 2^n$ for all $\alpha, \zeta \in \Re_+$, $\underline{\vartheta} \in \Re^n$, $n = 1, 2, \ldots$) The set $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{B}^n(\zeta, \underline{\vartheta}))$ consists of points (vectors) drawn from a one-dimensional space of $\Re^n$ described by the weight $\underline{w}$ and $\underline{w}\ \mathfrak{L}_{\underline{w}}(\alpha, \mathcal{B}^n(\zeta, \underline{\vartheta}))$, a collection of scalar products, is a discrete subset of $\Re$ in one-one correspondence with $\{1, 2, \ldots, 2^n\}$.

Preservance weights, ie, weights $\underline{w} \in \wp_n(\alpha)$, $\alpha \in \Re_+$, which establish a preservance of $\mathcal{B}^n(\zeta, \underline{\vartheta})$ in $\mathcal{L}_{\underline{w}}$ (through points in $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{B}^n(\zeta, \underline{\vartheta})) \subset \mathcal{L}_{\underline{w}}$), will in the following be enumerated with the notation $\underline{w}_{<\epsilon>}$ where ϵ denotes the enumerator index. Figure 3.2 (p. 122) indicates an enumeration scheme[8] for the preservance weights of the collection of scaled and translated binary vectors. As an example, all the preservance weights for an input space of dimensionality $n = 3$ (α chosen to be unity) are enumerated in Table 3.1 (p. 123). In the enumeration suggested above,

[8] The enumeration scheme is given in a pseudo code

```
for $\mathfrak{e} = 0, 1, 2, \ldots n!\,2^n - 1$ given $n$, $n = 1, 2, \ldots$; and $\alpha \in \Re_+$
do
    1 Evaluate $\mathfrak{e}_p = \left\lfloor \frac{\mathfrak{e}}{2^n} \right\rfloor$ and $\mathfrak{e}_s = \mathfrak{e} \,\mathbf{mod}\, 2^n$, where $\lfloor\ \rfloor$ and mod
      are the floor and modulo operations, respectively.
    2 Construct ${}_1\mathfrak{O} = \left\{ {}_1o_\imath = 2^{\imath-1} \,|\, \imath = 1, 2, \ldots n \right\}$, a set in which
      the elements are put in ascending order.
    3. Assign to $\imath$ the value 1 and to $e_\imath$ the value $\mathfrak{e}_p$
    4 Assign to $w_\imath$ the value ${}_\imath o_{\jmath_\imath}$, where $\jmath_\imath = \left\lfloor \frac{e_\imath}{(n-\imath)!} \right\rfloor$ and ${}_\imath o_{\jmath_\imath}$ is
      the $\overline{\jmath_\imath + 1}$th element of ${}_\imath\mathfrak{O}$
    5 Evaluate $e_{\imath+1} = e_\imath - (n-\imath)!\,\jmath_\imath$ and construct the ordered set
      ${}_{\imath+1}\mathfrak{O} = {}_\imath\mathfrak{O} \setminus \left\{ {}_\imath o_{\jmath_\imath} \right\}$, where the elements are in ascending
      order.
    6. Assign to $\imath'$ the value $\imath$, advance $\imath$ by 1 and repeat steps 4 and
      5 till $\imath'$ is not more than $n$
    7 Assign to numbers $s_k$, $k = 1, 2, \ldots n$, values +1 or −1 such
      that $\sum_{k=1}^{n} s_k 2^{k-1} = 2\mathfrak{e}_s - (2^n - 1)$ ($\underline{s} \triangleq [s_1, s_2, \ldots s_n]^{\mathsf{T}}$ is the
      binary representation – with symbols +1 and −1 – of the deci-
      mal number $\mathfrak{e}_s$.)
    8 Assign to $\underline{w}_{<\mathfrak{e}>}$ the direct product of vectors $\underline{s}$ scaled by $-\alpha$
      and $\underline{w} \triangleq [w_1, w_2, \ldots w_n]^{\mathsf{T}}$, ie, $w_{<\mathfrak{e}>_k} = -\alpha s_k w_k$, $k = 1, 2, \ldots n$,
      $\underline{w}_{<\mathfrak{e}>} \triangleq [w_{<\mathfrak{e}>_1}, w_{<\mathfrak{e}>_2}, \ldots w_{<\mathfrak{e}>_n}]^{\mathsf{T}}$.
done
```

Figure 3.2: Scheme for enumerating preservance weights of $\mathcal{B}^n(\zeta, \underline{\vartheta})$, $n = 1, 2, \ldots$;, $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$

Table 3.1: Preservance weights for $n = 3$ with $\alpha = 1$

$\underline{w}_{<0>} = [+1\ +2\ +4]$	$\underline{w}_{<1>} = [-1\ +2\ +4]$
$\underline{w}_{<2>} = [+1\ -2\ +4]$	$\underline{w}_{<3>} = [-1\ -2\ +4]$
$\underline{w}_{<4>} = [+1\ +2\ -4]$	$\underline{w}_{<5>} = [-1\ +2\ -4]$
$\underline{w}_{<6>} = [+1\ -2\ -4]$	$\underline{w}_{<7>} = [-1\ -2\ -4]$
$\underline{w}_{<8>} = [+1\ +4\ +2]$	$\underline{w}_{<9>} = [-1\ +4\ +2]$
$\underline{w}_{<10>} = [+1\ -4\ +2]$	$\underline{w}_{<11>} = [-1\ -4\ +2]$
$\underline{w}_{<12>} = [+1\ +4\ -2]$	$\underline{w}_{<13>} = [-1\ +4\ -2]$
$\underline{w}_{<14>} = [+1\ -4\ -2]$	$\underline{w}_{<15>} = [-1\ -4\ -2]$
$\underline{w}_{<16>} = [+2\ +1\ +4]$	$\underline{w}_{<17>} = [-2\ +1\ +4]$
$\underline{w}_{<18>} = [+2\ -1\ +4]$	$\underline{w}_{<19>} = [-2\ -1\ +4]$
$\underline{w}_{<20>} = [+2\ +1\ -4]$	$\underline{w}_{<21>} = [-2\ +1\ -4]$
$\underline{w}_{<22>} = [+2\ -1\ -4]$	$\underline{w}_{<23>} = [-2\ -1\ -4]$
$\underline{w}_{<24>} = [+2\ +4\ +1]$	$\underline{w}_{<25>} = [-2\ +4\ +1]$
$\underline{w}_{<26>} = [+2\ -4\ +1]$	$\underline{w}_{<27>} = [-2\ -4\ +1]$
$\underline{w}_{<28>} = [+2\ +4\ -1]$	$\underline{w}_{<29>} = [-2\ +4\ -1]$
$\underline{w}_{<30>} = [+2\ -4\ -1]$	$\underline{w}_{<31>} = [-2\ -4\ -1]$
$\underline{w}_{<32>} = [+4\ +1\ +2]$	$\underline{w}_{<33>} = [-4\ +1\ +2]$
$\underline{w}_{<34>} = [+4\ -1\ +2]$	$\underline{w}_{<35>} = [-4\ -1\ +2]$
$\underline{w}_{<36>} = [+4\ +1\ -2]$	$\underline{w}_{<37>} = [-4\ +1\ -2]$
$\underline{w}_{<38>} = [+4\ -1\ -2]$	$\underline{w}_{<39>} = [-4\ -1\ -2]$
$\underline{w}_{<40>} = [+4\ +2\ +1]$	$\underline{w}_{<41>} = [-4\ +2\ +1]$
$\underline{w}_{<42>} = [+4\ -2\ +1]$	$\underline{w}_{<43>} = [-4\ -2\ +1]$
$\underline{w}_{<44>} = [+4\ +2\ -1]$	$\underline{w}_{<45>} = [-4\ +2\ -1]$
$\underline{w}_{<46>} = [+4\ -2\ -1]$	$\underline{w}_{<47>} = [-4\ -2\ -1]$

the preservance weight $\underline{w}_{<0>}$ for all dimensions (n) and $\alpha = 1$ refers to the weight used in a positional representation of decimal numbers as n bit binary numbers.

Weights that preserve all points of the Boolean hyper-cube $\mathcal{B}^n(\zeta, \underline{\vartheta})$, $n = 1, 2, \quad ; \zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$, along a one-dimensional linear subspace (described by the weight) define a *generalized positional numbering system*. In Figure 3.1 (*p*. 114) the directions of weight vectors have been shown to have uniform angular spacing· this property though depicted in the case of two-dimensional vector collections is assured in higher dimensions by the corollary to Theorem 3.1.2. Seeking a structure to the collection of preservance weights, the following hold.

PROPOSITION **3.1.4** *Preservation of the Boolean hyper-cube* $\mathcal{B}^n(\zeta, \underline{\vartheta})$, $\zeta \in \Re_+$, $\underline{\vartheta} \in \Re^n$, *under a weight* $\underline{w}_{<\epsilon>} \in \wp_n(\alpha)$ *is equivalent to preservation under a permuted version of* $\underline{w}_{<0>} \in \wp_n(\alpha)$, *the permutation being given by* $\boldsymbol{P}_{\epsilon 0} = [p_{i,j}]_{i=1}^{n}{}_{j=1}^{n}$, *where,*

$$p_{i,j} = \begin{cases} sgn(w_{<\epsilon>_i}) & if\, j = 1 + log_2\, |w_{<\epsilon>_i}| \\ 0 & otherwise. \end{cases}$$

PROOF: Without any loss of generality, the statement will be established with $\alpha = 1$. Let the elements of $\underline{w}_{<\epsilon>}$ be expressed as

$$w_{<\epsilon>_i} = s_i 2^{o_i},\ s_i = \pm 1,\ o_i = 0, 1, \ \ldots n-1,\ i = 1, 2, \quad n,$$

in accordance with the prescription given in Theorem 3.1.1 (*p*. 113). Noting that $w_{<0>_i} = 2^{i-1}$ by choice and that $|w_{<\epsilon>_i}| \neq |w_{<\epsilon>_j}|$, $i \neq j$,

$i, j = 1, 2, \ldots n$, the assignment suggested for the elements $p_{i,j}$ immediately follows on recognizing that $o_i = log_2 \; |w_{<\iota>,i}|$ and $s_i = sgn(w_{<\iota>,i})$.

□

Corollary to Proposition 3.1.4

a. *The permutation of* $\underline{w}_{<\iota>}$ *to* $\underline{w}_{<0>}$ *is given by* $\boldsymbol{P}_{0\iota} = \boldsymbol{P}_{\iota 0}^{\top}$, *where,* $\boldsymbol{A}^{\top}$ *denotes the transpose of a matrix* $\boldsymbol{A}$.

b. *The permutation of* $\underline{w}_{<\iota_1>}$ *to* $\underline{w}_{<\iota_2>}$ *is given by* $\boldsymbol{P}_{\iota_1 \iota_2} = \boldsymbol{P}_{\iota_2 0} \boldsymbol{P}_{0 \iota_1}$.

This statement is obvious and, hence, no proof is provided. In view of Proposition 3.1.4 (p. 124) and its corollary, a characterization of processor representation in isolated neurons does not lose generality when the weights are chosen to be in any of the finitely many directions suggested by the corollary to Theorem 3.1.2 (p. 115). Proposition 3 1.3 (p. 120) indicates that the preservance weight of a scaled and translated Boolean hyper-cube belongs to a scaled Boolean hyper-cube, though as indicated in Theorem 3.1.2 (p. 115), not all points of a (scaled) Boolean hyper-cube are valid as preservance weights of Boolean hyper-cubes.

It is of interest to know the possibility of using valid members of the scaled and translated Boolean hyper-cube[9] $\mathcal{B}^n(\zeta_1, \underline{\vartheta}_1)$ as the preservance weights of $\mathcal{B}^n(\zeta_2, \underline{\vartheta}_2)$, a Boolean hyper-cube with a scale factor

[9] Note that only a few members of any scaled and translated Boolean hyper-cube contribute to the collection of preservance weight $\wp_n$.

and translation allowed to be distinct from that of the Boolean hypercube from which the preservance weights are chosen. Referring to Equation 3.1a (*p* 110) the evaluation of η at an input $\underline{x}$ under a valid preservance weight $\underline{w} \in \mathcal{B}^n(\zeta_1, \underline{\vartheta}_1)$ – the preservance weight is assumed, for simplicity, to be described as $\underline{w} = \zeta_1 \underline{w}_{<0>} + \underline{\vartheta}_1$, $\underline{w}_{<0>} \in \wp_n(1)$ – is given as $\eta(\underline{x}) = \zeta_1 \underline{w}_{<0>}\ \underline{x} + \underline{\vartheta}_1\ \underline{x} - \theta$.

Theorem 3.1 2 (*p*. 115) shows that preservance is unaffected by scaling in the weights, the scale factor is assumed positive. As a result only the effect of superposition of weight vectors on preservance remains to be studied. The structure of η for weights in $\mathcal{B}^n(\zeta_1, \underline{\vartheta}_1)$ suggests two possibilities: (a) the composition of preservance weights through weights that are not themselves valid as preservance weights and (b) the decomposition of preservance weights in terms of preservance weights. Of these only the latter situation is of interest as the former is trivially satisfied by the structure of vector spaces. The following statement establishes a characterization of preservance weights.

$\mathcal{THEOREM}$ **3.1.3** *Given two preservance weights of the discrete space* $\mathcal{B}^n(\zeta, \underline{\vartheta})$, $\zeta \in \Re_+$, $\underline{\vartheta} \in \Re^n$, $\underline{w}_1 \in \wp_n(\alpha_1)$ *and* $\underline{w}_2 \in \wp_n(\alpha_2)$, $\alpha_1, \alpha_2 \in \Re_+$, *a weight* $\underline{w}$ *given as* $\underline{w} = \underline{w}_1 + \underline{w}_2$ *is a preservance weight of* $\mathcal{B}^n(\zeta, \underline{\vartheta})$ *if and only if*

$$\frac{|\underline{w}_1\ \underline{w}_2|}{\|\underline{w}_1\|\ \|\underline{w}_2\|} = 1,$$

with $\alpha_1 \neq \beta\alpha_2$ *if* $\underline{w}_2 = -\beta\underline{w}_1$.

Proof If $|\underline{w}_1 \cdot \underline{w}_2|$ is equal to $||\underline{w}_1||\,||\underline{w}_1||$ then $\underline{w}_2 \in \mathcal{L}_{\underline{w}_1}$ which implies that $\underline{w} \in \mathcal{L}_{\underline{w}_1}$. As a consequence of Theorem 3.1.2 (*p.* 115) all members of the one-dimensional subspace (of $\Re^n$) $\mathcal{L}_{\underline{w}_1} \setminus \{\underline{0}\}$ are preservance weights of the discrete set $\mathcal{B}^n(\zeta, \underline{\vartheta})$ if the weight $\underline{w}_1$ is a preservance weight of $\mathcal{B}^n(\zeta, \underline{\vartheta})$. $\alpha_1 \neq \beta\alpha_2$ if $\underline{w}_2 = -\beta\underline{w}_1$ ensures that the composition $\underline{w}$ is not a null vector. This establishes the 'if' part.

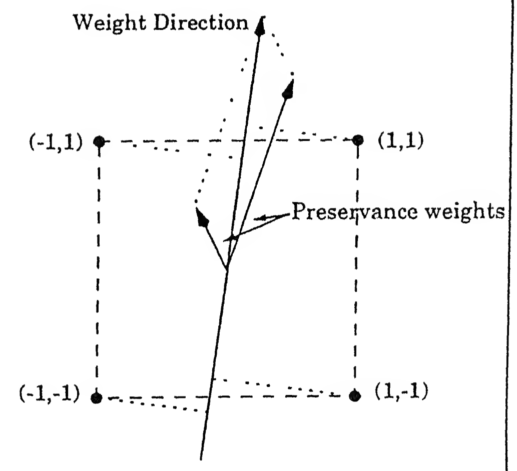


Figure 3.3: Superposition of preservance weights of $\mathcal{B}^2$

Figure 3 3 illustrates a weight vector composed from preservance weights that do not satisfy the requirement $|\underline{w}_1 \cdot \underline{w}_2| = ||\underline{w}_1||\,||\underline{w}_2||$. In such a superposition, either one-one correspondence or regularity in the images of $\mathcal{B}^n(\zeta, \underline{\vartheta})$ are not exhibited and, as a consequence, the composition $\underline{w}$ fails to be a preservance weight for $\mathcal{B}^n(\zeta, \underline{\vartheta})$. This observation counters the negation of the 'only if' part and the resulting contradiction establishes the necessity of the constituent preservance weights in a composition to be in the same 'direction.'

□

This theorem states that a superposition of the preservance weights of $\mathcal{B}^n(\zeta,\underline{\vartheta})$ is a preservance weight of $\mathcal{B}^n(\zeta,\underline{\vartheta})$ if and only if the constituent preservance weights differ from each other only in scale. As a consequence, preservance weights of the discrete space $\mathcal{B}^n(\zeta,\underline{\vartheta})$ cannot be selected as basis vectors of $\wp_n$. (Given that the finite number of directions in which preservance weights can be chosen is exponentially dependent on the dimensionality n and the structure of preservance weights as given in the corollary to Theorem 3.1 2 (*p*. 115) it is not difficult to find a collection of n linearly independent preservance weights.)

As established in the preceding discussion, the operation of inner product employing preservance weights induces a one-one correspondence between the 2^n vertices (*ie*, points) of $\mathcal{B}^n(\zeta,\underline{\vartheta})$ and the 2^n uniformly spaced points in the one-dimensional sub-space $\mathcal{L}_{\underline{w}}(\alpha,\mathcal{B}^n(\zeta,\underline{\vartheta})) \subset \mathcal{L}_{\underline{w}}$, for any $\alpha \in \Re_+$. Under this operation, it is easy to see that every point in $\mathcal{L}_{\underline{w}}$ is identified with a distinct hyper-plane in $\Re^n$. Preservation of points in an input space then amounts to an identification of specific hyper-planes and choice of a specific point in each hyper-plane. These specific points are now identified with distinct points in the one-dimensional sub-space identified by the preservance weight.

The discrete space $\mathcal{B}^n$ is a collection of input vectors whose elements are commonly interpreted as an assertion of the presence, or absence, of features related to observations that are subjected to inferencing (In this sense **McCulloch & Pitts** (1943) and subsequent investigators have identified neuron inputs with propositions and the operation of neurons (as well as neural networks) with formulae of the propositional calculus) A generalization of $\mathcal{B}^n(\zeta, \underline{\vartheta})$ to discrete subsets of $\Re^n$ that are chosen to have a regular structure and are in one-one correspondence with discrete subsets of $\mathcal{L}_{\underline{w}}$ for a given preservance weight $\underline{w}$ is provided in the following.

Consider the construction for a given n, $n = 1, 2, . \; .,$

$$\mathcal{S}^n_r(\zeta, \underline{\vartheta}) = \bigcup_{\imath=1}^{r} \mathcal{B}^n(\zeta_\imath, \underline{\vartheta}),$$

where, $r = 1, 2, \quad , \underline{\vartheta} \in \Re^n$ and the coefficients $\zeta_\imath$ and $\zeta_\jmath$, $\zeta_\imath, \zeta_\jmath \in \Re_+$, are relatively (mutually) prime for all $\imath, j, i \neq j, \imath, \jmath = 1, 2, .. \; , r$. This space is constructed as a union of scaled Boolean hyper-cubes with a common translation taking care that the scale factors do not force multiple points to have the same image in $\mathcal{L}_{\underline{w}}$ for a preservance weight $\underline{w}$ corresponding to the constituent discrete sets $\mathcal{B}^n(\zeta_i, \underline{\vartheta})$. It is of ınterest now to seek the preservance weights of $\mathcal{S}^n_r(\zeta, \underline{\vartheta})$ based on the available knowledge about the preservance weights of scaled and translated Boolean hyper-cubes.

Proposition 3.1.5 *Weights $\underline{w} \in \wp_n$ establish distınct images, in $\mathcal{L}_{\underline{w}}$, of all points of the dıscrete space $\mathcal{S}^n_r(\zeta, \underline{\vartheta})$ for all $n = 1, 2, . \quad ; r = 1, 2, \ldots;$ $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$.*

Proof: Every Boolean hyper-cube $\mathcal{B}^n(\zeta_\imath, \underline{\vartheta}) \subset \mathcal{S}^n_r(\zeta, \underline{\vartheta})$, $\zeta_\imath \in \Re_+$ is preserved in $\mathcal{L}_{\underline{w}}$ as indicated in Proposition 3.1.3 (*p*. 120). When these hyper-cubes are scaled using coefficients that are mutually prime, the preservation points in $\mathcal{L}_{\underline{w}}$ corresponding to the different hyper-cubes are distinct and, hence, distinctness of the images in the union is assured. On the other hand, if the coefficients corresponding to different hyper-cubes are not relatively prime, the required one-one correspondence between $\mathcal{S}_r^n(\zeta, \underline{\vartheta})$ and any discrete subset of $\mathcal{L}_{\underline{w}}$ cannot be ensured.

□

Figure 3.4 provides an illustration of the discrete space $\mathcal{S}_3^2(1, \underline{0})$ with an accompanying diagram of points in $\mathcal{L}_{\underline{w}_{<0>}}$ that are in one-one correspondence with the points in the discrete space. As evident from this illustration, preservation points in $\mathcal{L}_{\underline{w}}$ (for a preservance weight $\underline{w}$) are, in general, irregularly spaced, the spacing increasing as the distance from the preservation point corresponding to the centroid of $\mathcal{S}_r^n(\zeta, \underline{\vartheta})$. The lack of uniformity in spacing between the points identified in $\mathcal{L}_{\underline{w}}$ corresponding to points in $\mathcal{S}_r^n(\zeta, \underline{\vartheta})$ given the scaling coefficients $\zeta_\imath$, $\imath = 1, 2, \ \ ., r$, precludes any further consideration of preservation of the discrete space $\mathcal{S}_r^n(\zeta, \underline{\vartheta})$. However, the approach of identifying points in $\mathcal{L}_{\underline{w}}$ with a union of scaled and (origin) translated versions of the basic Boolean hyper-cube $\mathcal{B}^n(1, \underline{0})$ is useful as indicated in the following.

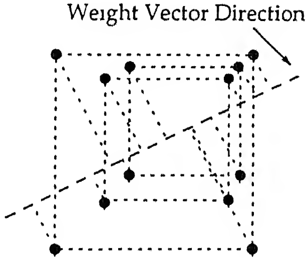


Figure 3.4: $\mathcal{S}_3^2(1, \underline{0})$

Let $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ denote the recursive construction

$$\mathcal{P}_0^n(\zeta,\underline{\vartheta}) = \emptyset, \tag{3.3a}$$

$$\mathcal{P}_1^n(\zeta,\underline{\vartheta}) = \mathcal{B}^n(\zeta,\underline{\vartheta}) \tag{3 3b}$$

$$\mathcal{P}_r^n(\zeta,\underline{\vartheta}) = \mathcal{P}_{r-1}^n(\zeta,\underline{\vartheta})\bigcup\left(\bigcup_{i=1}^{2^{nr}}\mathcal{B}^n(2^{-(r-1)n}\zeta,\underline{\vartheta}_i^{(r)}+\underline{\vartheta})\right), \quad r = 2,3,\ldots, \tag{3.3c}$$

where, $\underline{\vartheta}_i^{(r)}$, $i = 1,2,\ldots,2^{nr}$, are the 2^{nr} (ordered) points of the discrete subset $\mathcal{P}_{r-1}^n(\zeta,\underline{\vartheta})\setminus\mathcal{P}_{r-2}^n(\zeta,\underline{\vartheta})$ of $\Re^n$, $n = 1,2,\ldots;$ $r = 1,2,\ldots;$ $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$. This discrete set differs from $\mathcal{S}_r^n(\zeta,\underline{\vartheta})$ in that the constituent Boolean hyper-cubes are scaled and translated, the scale factors and translations being related in powers of 2. Seeking the preservance of $\mathcal{P}^n(\zeta,\underline{\vartheta})$ the following hold.

PROPOSITION **3.1.6** *Weights $\underline{w} \in \wp_n$ also preserve, in $\mathcal{L}_{\underline{w}}$, all points of $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, $n = 1,2,\ldots$, $r = 1,2,\ldots$, $\zeta \in \Re_+$, and $\underline{\vartheta} \in \Re^n$.*

PROOF: For any weight $\underline{w} \in \wp_n$ every scaled and translated Boolean hyper-cube $\mathcal{B}^n(2^{-(r-1)n}\zeta,\underline{\vartheta}_i^{(r)}+\underline{\vartheta}) \subset \mathcal{P}_r^n(\zeta,\underline{\vartheta})$, for all values of r, $r = 1,2,\ldots$, is preserved in $\mathcal{L}_{\underline{w}}$ as established in Proposition 3.1.3 (*p.* 120). Thus preservance of $\mathcal{P}_1^n(\zeta,\underline{\vartheta})$ is ensured. At any step r, $r = 2,3,\ldots$, the images in $\mathcal{L}_{\underline{w}}$ of points in the union of Boolean hyper-cubes that specify points in addition to those accumulated at the end of step $r-1$ are uniformly spaced, with the spacing between adjacent points being given as 2^{-n} times the smallest spacing between images in $\mathcal{L}_w$ of points in

$\mathcal{P}^n_{r-1}(\zeta,\underline{\vartheta})$. (See Figure 3 5 ($p$. 134) for a clarification of the construction when $n = 2$ and $r = 3$.) As the translations in step r are given by the points added in step $r-1$, the collection of uniformly spaced images in $\mathcal{L}_{\underline{w}}$ are distinct in all the steps, thereby satisfying one-one correspondence and regularity. Order preservation, based on the preservance of $\mathcal{B}^n(\zeta,\underline{\vartheta})$, follows from the fact that the set $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ and its image in $\mathcal{L}_{\underline{w}}$ are composed of non-overlapping sets, the constituent sets being, respectively, scaled and translated Boolean hyper-cubes and their images in $\mathcal{L}_{\underline{w}}$.

□

PROPOSITION 3.1.7 *The centroid of $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ is identical to that of $\mathcal{B}^n(\zeta,\underline{\vartheta})$ $n = 1, 2, \ldots, r$, $r = 1, 2, \ldots$, $\zeta \in \Re_+$, and $\underline{\vartheta} \in \Re^n$*

Proposition 3.1.7 is obvious. In view of Proposition 3.1.6 $\wp_n$, the space of preservance weights, will be referred to, in the sequel, as the collection of weights that preserve $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ in $\mathcal{L}_{\underline{w}}$. The collection of points in $\mathcal{L}_{\underline{w}}$ that are put in one-one correspondence with $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ by the preservance weights will be denoted by $\mathfrak{L}_{\underline{w}}(\alpha,\mathcal{P}^n_r(\zeta,\underline{\vartheta}))$. (Clearly, $\mathfrak{L}_{\underline{w}}(\alpha,\mathcal{P}^n_r(\zeta,\underline{\vartheta})) \supset \mathfrak{L}_{\underline{w}}(\alpha,\mathcal{B}^n(\zeta,\underline{\vartheta}))$, for all $\alpha,\zeta \in \Re_+$, $\underline{\vartheta} \in \Re^n$, and the containment is proper when $r = 2, 3, \ldots$) Note that as a consequence of the manner in which the space $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ is constructed, the interval between any two adjacent preservation points of $\mathcal{P}^n_{r-1}(\zeta,\underline{\vartheta})$ in $\mathcal{L}_{\underline{w}}$ is sub-divided to accommodate 2^n additional preservation points. This amounts to a ranking (denoted by r) of the preservation points in $\mathcal{L}_{\underline{w}}$.

An illustration of the discrete space $\mathcal{P}_3^2(1,\underline{0})$ with an accompanying diagram of points in $\mathcal{L}_{\underline{w}_{<0>}}$ that are in isomorphism with the points in the discrete space is provided in Figure 3.5: the points of $\Re^2$ that belong to $\mathcal{P}_3^2(1,\underline{0})$ and are in one-one correspondence (denoted by ↔) with those of $\mathfrak{L}_{\underline{w}_{<0>}}(1,\mathcal{P}_r^n(1,\underline{0}))$ (corresponding to the preservance weight $w_{<0>}$ with $\alpha = 1$) are indicated in Table 3.2 (*p.* 135). From the illustration, regularity of the discrete space and uniformity of spacing between preservation points in $\mathcal{L}_{\underline{w}}$ are immediately evident. The following characteristics of $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ are noteworthy.

Proposition 3.1.8 *The number of distinct points of $\Re^n$ included in the discrete space $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ is given by*

$$\begin{aligned} |\mathcal{P}_r^n(\zeta,\underline{\vartheta})| &= 2^n\left(|\mathcal{P}_{r-1}^n(\zeta,\underline{\vartheta})| - |\mathcal{P}_{r-2}^n(\zeta,\underline{\vartheta})|\right) + |\mathcal{P}_{r-1}^n(\zeta,\underline{\vartheta})|, \\ &\equiv \frac{2^n\left(2^{nr}-1\right)}{2^n-1}, \; r = 2, 3, \ldots, \end{aligned}$$

subject to the understanding that $|\mathcal{P}_0^n(\zeta,\underline{\vartheta})| = 0$ and $|\mathcal{P}_1^n(\zeta,\underline{\vartheta})| = 2^n$, $n = 1, 2, \ldots$, $\zeta \in \Re_+$, and $\underline{\vartheta} \in \Re^n$.

Proof: For all n, $n = 1, 2, \ldots$, the construction indicated in Equation 3.3 (*p.* 131) suggests that $\mathcal{P}_1^n(\zeta,\underline{\vartheta})$ is a Boolean hyper-cube and thus $|\mathcal{P}_1^n(\zeta,\underline{\vartheta})| = 2^n$ for all the admissible values of ζ and $\underline{\vartheta}$. The space $\mathcal{P}_2^n(\zeta,\underline{\vartheta})$ is obtained by identifying with every vertex of $\mathcal{P}_1^n(\zeta,\underline{\vartheta})$ a discrete subspace of $\Re^n$ isomorphic to $\mathcal{P}_1^n(\zeta,\underline{\vartheta})$ and this, on taking unions of the discrete subspaces, establishes that $|\mathcal{P}_2^n(\zeta,\underline{\vartheta})| = 2^n\,|\mathcal{P}_1^n(\zeta,\underline{\vartheta})| + |\mathcal{P}_1^n(\zeta,\underline{\vartheta})| \equiv 2^{2n} + 2^n$.

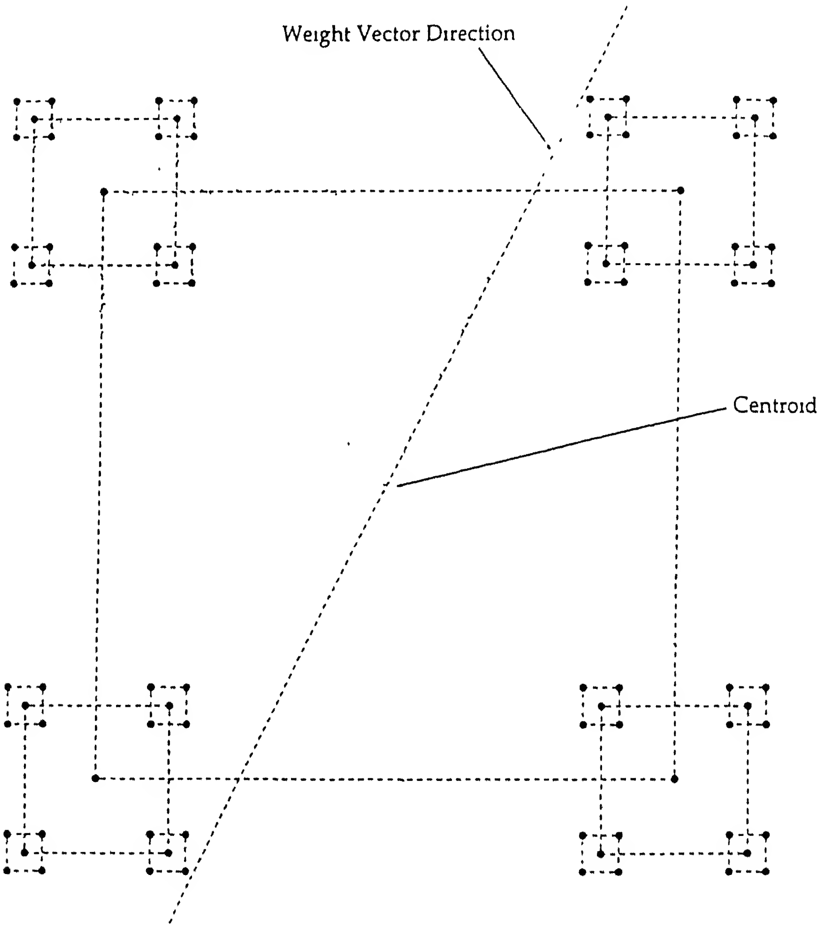


Figure 3.5: Illustration of $\mathcal{P}_3^2(1.\underline{0})$

Table 3.2: Points of $\mathcal{P}_3^2(1,\underline{0}) \leftrightarrow \frac{1}{\|\underline{w}_{<0>}\|^2}\underline{w}^{\mathrm{T}}_{<0>}\mathfrak{L}_{\underline{w}_{<0>}}(1,\mathcal{P}_3^2(1,\underline{0}))$

(-1 3125,-1 3125)	↔	-3 9375	(-1 3125,-1 1875)	↔	-3 8125	(-1 25,-1 25)	↔	-3 75
(-1 1875,-1 3125)	↔	-3 6875	(-1.1875,-1 1875)	↔	-3 5625	(-1 3125,-0 8125)	↔	-3 4375
(-1 3125,-0 6875)	↔	-3 3125	(-1 25,-0 75)	↔	-3 25	(-1 1875,-0 8125)	↔	-3 1875
(-1 1875,-0 6875)	↔	-3 0625	(-1,-1)	↔	-3	(-0 8125,-1 3125)	↔	-2 9375
(-0 8125,-1 1875)	↔	-2.8125	(-0 75,-1 25)	↔	-2 75	(-0 6875,-1 3125)	↔	-2 6875
(-0 6875,-1 1875)	↔	-2.5625	(-0 8125,-0 8125)	↔	-2 4375	(-0 8125,-0 6875)	↔	-2 3125
(-0 75,-0 75)	↔	-2 25	(-0 6875,-0 8125)	↔	-2 1875	(-0 6875,-0 6875)	↔	-2 0625
(-1 3125,0 6875)	↔	-1 9375	(-1 3125,0 8125)	↔	-1 8125	(-1 25,0 75)	↔	-1 75
(-1 1875,0 6875)	↔	-1 6875	(-1.1875,0 8125)	↔	-1 5625	(-1 3125,1 1875)	↔	-1 4375
(-1 3125,1 3125)	↔	-1 3125	(-1 25,1 25)	↔	-1 25	(-1 1875,1 1875)	↔	-1 1875
(-1 1875,1 3125)	↔	-1 0625	(-1,1)	↔	-1	(-0 8125,0 6875)	↔	-0 9375
(-0 8125,0 8125)	↔	-0 8125	(-0.75,0 75)	↔	-0 75	(-0 6875,0 6875)	↔	-0.6875
(-0.6875,0 8125)	↔	-0 5625	(-0 8125,1 1875)	↔	-0 4375	(-0 8125,1 3125)	↔	-0 3125
(-0 75,1 25)	↔	-0 25	(-0 6875,1 1875)	↔	-0 1875	(-0 6875,1 3125)	↔	-0 0625
(0 6875,-1 3125)	↔	0 0625	(0 6875,-1 1875)	↔	0 1875	(0 75,-1 25)	↔	0 25
(0 8125,-1 3125)	↔	0.3125	(0 8125,-1.1875)	↔	0 4375	(0 6875,-0 8125)	↔	0 5625
(0 6875,-0 6875)	↔	0 6875	(0 75,-0 75)	↔	0 75	(0 8125,-0 8125)	↔	0 8125
(0 8125,-0 6875)	↔	0 9375	(1,-1)	↔	1	(1 1875,-1 3125)	↔	1 0625
(1 1875,-1.1875)	↔	1 1875	(1 25,-1 25)	↔	1 25	(1 3125,-1 3125)	↔	1 3125
(1 3125,-1.1875)	↔	1 4375	(1 1875,-0 8125)	↔	1 5625	(1 1875,-0 6875)	↔	1 6875
(1 25,-0 75)	↔	1 75	(1.3125,-0 8125)	↔	1 8125	(1 3125,-0 6875)	↔	1 9375
(0 6875,0 6875)	↔	2 0625	(0 6875,0 8125)	↔	2 1875	(0 75,0 75)	↔	2 25
(0.8125,0.6875)	↔	2 3125	(0 8125,0 8125)	↔	2.4375	(0 6875,1 1875)	↔	2 5625
(0 6875,1 3125)	↔	2 6875	(0.75,1 25)	↔	2 75	(0 8125,1 1875)	↔	2 8125
(0 8125,1 3125)	↔	2 9375	(1,1)	↔	3	(1 1875,0 6875)	↔	3 0625
(1 1875,0 8125)	↔	3 1875	(1 25,0 75)	↔	3.25	(1 3125,0 6875)	↔	3 3125
(1 3125,0 8125)	↔	3 4375	(1.1875,1.1875)	↔	3 5625	(1 1875,1 3125)	↔	3 6875
(1 25,1.25)	↔	3 75	(1 3125,1.1875)	↔	3.8125	(1.3125,1 3125)	↔	3 9375

In the construction of spaces $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, $r = 3, 4,$, association of $\mathcal{P}_1^n(\zeta,\underline{\vartheta})$ is, however, not with respect to every point of $\mathcal{P}_{r-1}^n(\zeta,\underline{\vartheta})$ but only with the points in $\mathcal{P}_{r-1}^n(\zeta,\underline{\vartheta})\backslash\mathcal{P}_{r-2}^n(\zeta,\underline{\vartheta})$, *ie*, points of $\mathcal{P}_{r-1}^n(\zeta,\underline{\vartheta})$ not contained in $\mathcal{P}_{r-2}^n(\zeta,\underline{\vartheta})$. This establishes the recursive relationship in the cardinality of the spaces $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, $r = 2, 3,$, which gets extended to the case when $i = 1$ with the assumption that $|\mathcal{P}_0^n(\zeta,\underline{\vartheta})| = 0$. (This assumption is justified as $\mathcal{P}_0^n(\zeta,\underline{\vartheta})$ is defined to be the empty set) On carrying out the suggested recursion, the cardinality of $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ is obtained as the geometric series:

$$|\mathcal{P}_r^n(\zeta,\underline{\vartheta})| = \sum_{i=1}^{r} 2^{in}$$

It is then simple to see that $|\mathcal{P}_r^n(\zeta,\underline{\vartheta})|$ is indeed given by $\frac{2^n\,(2^{nr}-1)}{2^n-1}$.

□

Note that this result follows directly from the definition as in the rth step of the recursion, 2^{nr} points are being added. However, a more detailed argument has been provided in the proof to help a clarification of the construction of the discrete space $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$. Table 3.3 lists the value of $|\mathcal{P}_r^n(\zeta,\vartheta)|$ for a few small values of n and r. Continuing the recursion to the limit, the following emerges.

PROPOSITION 3.1.9 $\mathcal{P}_r^n(\zeta,\underline{\vartheta}) \subset \Re^n$, $n = 1, 2, .\ .$; $r = 1, 2, \ldots$; $\zeta \in \Re_+$ *and* $\underline{\vartheta} \in \Re^n$, *is a discrete space containing points sampled from open balls centered at the vertices of the Boolean hyper-cube* $\mathcal{B}^n(\zeta,\underline{\vartheta})$. *The radius of each of the* 2^n *distinct balls is given by* $\lim_{r\to\infty} \frac{2^{-n}(2^{-nr}-1)}{(2^{-n}-1)}$.

Table 3.3: Cardinality of $\mathcal{P}_r^n(1,\underline{0})$

n	r = 1	r = 2	r = 3	r = 4	r = 5
1	2	6	14	30	62
2	4	20	84	340	1364
3	8	72	584	4680	37,448
4	16	272	4368	69,904	1,118,480
5	32	1056	33,824	1,082,400	34,636,832
6	64	4160	266,304	17,043,520	1,090,785,344
7	128	16,512	2,113,664	270,549,120	34,630,287,488
8	256	65,792	16,843,008	4,311,810,304	1,103,823,438,080
9	512	262,656	134,480,384	68,853,957,120	35,253,226,045,952
10	1024	1,049,600	1,074,791,424	1,100,586,419,200	1,127,000,493,261,824

PROPOSITION **3.1.10** *The collection of preservation points of the discrete space $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, $n = 1,2,\ldots$, $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$, as r and ζ tend to ∞, given a preservance weight $\underline{w}$, is dense in $\mathcal{L}_{\underline{w}}$.*

This latter statement implies that the collection of hyper-planes that are identified by the preservation points of $\mathcal{L}_{\underline{w}}$, given a preservance weight $\underline{w}$, is dense in the input space $\Re^n$. For any finite positive value of ζ, however, only finitely many preservation points exist in a bounded interval of $\mathcal{L}_{\underline{w}}$ for a preservance weight $\underline{w}$. As indicated in Table 3.3 the number of points in $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ grows faster (with the dimensionality n) than those in $\mathcal{B}^n(\zeta,\underline{\vartheta}) \equiv \mathcal{P}_1^n(\zeta,\underline{\vartheta})$ and, thereby, a comparatively larger number of functions are realized even with small values of r: in this comparison the functions are assumed to have discrete outputs.

3.2 Function Representation in Isolated Neurons

Existence of weights $\underline{w}$ that preserve all points of $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ in $\mathcal{L}_{\underline{w}}$ for the admissible values of n, r, ζ and $\underline{\vartheta}$ ensures that functions defined over $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ through the operation introduced in Equation 3.1 (p 110) can be appreciated as univariate functions when preservance weights are employed in evaluating the projections of input patterns $\underline{x}$. While it is trivial to note that the operation of inner product maps $\Re^n$ to $\Re$, selection of preservance weights enables a parameterized description of the (discrete) input space: the projection of input points along the preservance weight is used as the parameter.

In order that the equivalent description of functions over the discrete space $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ be understood it is important to note that functions over $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ when realized through isolated neurons incorporating a weight $\underline{w}$ in $\wp_n(\alpha)$, $\alpha \in \Re_+$, are equivalently realized as functions over $\mathfrak{L}_{\underline{w}}(\alpha,\mathcal{P}_r^n(\zeta,\underline{\vartheta}))$. Note that $\mathfrak{L}_{\underline{w}}(\alpha,\mathcal{P}_r^n(\zeta,\underline{\vartheta}))$ is a one-dimensional discrete subset of $\Re^n$ and contains vectors which are in $\mathcal{L}_{\underline{w}}$. However, $\frac{1}{\|\underline{w}\|^2}\underline{w}\cdot\mathfrak{L}_{\underline{w}}(\alpha,\mathcal{P}_r^n(\zeta,\underline{\vartheta}))$, the collection of normalized projections along the preservance weight is in one-one correspondence with $\{1,2,\ldots|\mathcal{P}_r^n(\zeta,\underline{\vartheta})|\}$. These observations are formalized below.

$\mathcal{PROPOSITION}$ **3.2.1** *Functions defined over the discrete space $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, $n = 1,2,\ldots$; $r = 1,2,\ldots$; $\zeta \in \Re_+$, $\underline{\vartheta} \in \Re^n$, are equivalent, under inner product employing a preservance weight $\underline{w} \in \wp_n(\alpha)$, for any $\alpha \in \Re_+$, to (finite length) sequences over the discrete index set $\mathfrak{L}_{\underline{w}}(\alpha,\mathcal{P}_r^n(\zeta,\underline{\vartheta}))$.*

Corollary to Proposition **3.2.1** *Boolean functions,* ie, *binary functions over $\mathcal{P}^n_1(1,\underline{0})$, $n = 1, 2, .\ \ ,$ under inner product incorporating a preservance weight $\underline{w} \in \wp_n(\alpha)$, for any $\alpha \in \Re_+$, are equivalent to (finite length) binary sequences over the discrete index set $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}^n_r(\zeta, \underline{\vartheta}))$.*

Proposition **3.2.2** *Decision functions evaluated by neurons with preservance weights $\underline{w} \in \wp_n$ are*

1. *piece-wise constant (bivalent) sequences over $\mathcal{L}_{\underline{w}} \subset \Re^n$ if the activation function σ is hard-limiting and*
2. *continuous and piece-wise monotonic sequences over $\mathcal{L}_{\underline{w}}$ if σ is sigmoidal.*

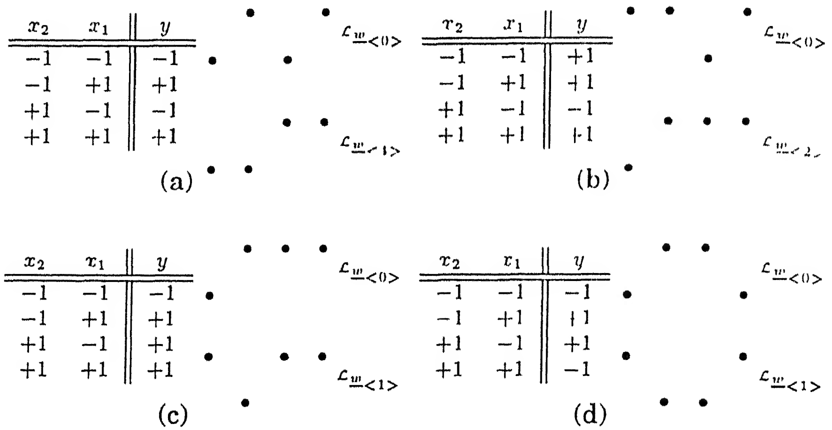
Theorem **3.2.1** *The number of distinct k-ary functions that can be defined on $\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ is*

$$k^{|\mathcal{P}^n_r(\zeta,\underline{\vartheta})|} \equiv k^{2^n(2^{nr}-1)/(2^n-1)}, \; n, r = 1, 2, \ ..; \; \zeta \in \Re_+ \textit{ and } \underline{\vartheta} \in \Re^n$$

and tends to $k^{2^{nr}}$ as n increases towards ∞.

Corollary to Theorem **3.2.1** *The number of distinct binary valued functions that can be defined on $\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ is $2^{2^n(2^{nr}-1)/(2^n-1)}$.*

A simple application of combinatorial arguments using the cardinality of $\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ (see Proposition 3.1.8 (*p.* 133)) is sufficient to establish the above theorem and its corollary. Though a discussion on


$\mathcal{L}_{\underline{w}<0>}$
$\mathcal{L}_{\underline{w}<1>}$
$\mathcal{L}_{\underline{w}<0>}$
$\mathcal{L}_{\underline{w}<2>}$
$\mathcal{L}_{\underline{w}<0>}$
$\mathcal{L}_{\underline{w}<1>}$
$\mathcal{L}_{\underline{w}<0>}$
$\mathcal{L}_{\underline{w}<1>}$

x_2	x_1	y
-1	-1	-1
-1	$+1$	$+1$
$+1$	-1	-1
$+1$	$+1$	$+1$

(a)

x_2	x_1	y
-1	-1	$+1$
-1	$+1$	$+1$
$+1$	-1	-1
$+1$	$+1$	$+1$

(b)

x_2	x_1	y
-1	-1	-1
-1	$+1$	$+1$
$+1$	-1	$+1$
$+1$	$+1$	$+1$

(c)

x_2	x_1	y
-1	-1	-1
-1	$+1$	$+1$
$+1$	-1	$+1$
$+1$	$+1$	-1

(d)

Figure 3.6: Representation of bipolar bivalent functions on $\mathcal{P}_1^2(1,\underline{0})$

the nature of functions represented in isolated neurons can effortlessly be carried out on multi-valued discrete functions–in particular when preservance weights are incorporated in Equation 3.1a (*p*. 110)–only the specific case of bivalent functions, *ie*, functions which take on one of two distinct values, will be considered: such functions are important in categorization, specially in the construction of dichotomies.

Assuming that the activation function σ is bipolar and is symmetric about the origin in the range, *ie*, $\zeta_+ + \zeta_- = 0$, where ζ_+ and ζ_- are the extreme values taken on by σ, as indicated in Chapter 2, it is easy to see that every function on $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, for each of the admissible values of n, r, ζ and $\underline{\vartheta}$, is characterized, largely, by the number of sign changes in the sequence defined on $\mathcal{L}_{\underline{w}}(\alpha,\mathcal{P}_r^n(\zeta,\underline{\vartheta}))$. Figure 3.6 (*p*. 140) presents some examples of (bipolar) bivalent sequences (binary if a hard limiting activation function is in use) representing (bipolar) bivalent functions on $\mathcal{P}_1^2(1,\underline{0})$. These examples show the following.

$\mathfrak{P}$ROPOSITION 3.2.3 *The number of distinct sign changes in the sequences over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta,\underline{\vartheta}))$ which represent (bipolar bivalent) functions over $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, under a preservance weight $\underline{w} \in \wp_n$, for any $\alpha \in \Re_+$, varies from a minimum of* 0 *to a maximum of* $|\mathcal{P}_r^n(\zeta,\underline{\vartheta})| - 1$.

Recalling the operational nature of an isolated neuron, as detailed in Equation 3.1, the output y, as a function of the input pattern $\underline{x}$, is realized as the effect of a transformation σ on the inner product $\underline{w}\cdot\underline{x}$, the latter being shifted (in range) by the threshold θ. In the context of pattern recognition and processor realization it is of interest to know the nature of grouping in the collection of pre-images of the distinct labels (or types of labels) that get assigned to the function. An investigation of the discrete sequences that correspond to neuron outputs defined over $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ leads to the following.

$\mathfrak{P}$ROPOSITION 3.2.4 *Neurons with activation functions that are sigmoidal (including hard-limiter) and a preservance weight $\underline{w} \in \wp_n(\alpha)$, for any $\alpha \in \Re_+$, induce no more than one sign transition in the (finite length) sequence over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta,\underline{\vartheta}))$ representing functions over $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, $n = 1, 2, \ldots$, $r = 1, 2, \ldots$, $\zeta \in \Re_+$, $\underline{\vartheta} \in \Re^n$.*

The above statement, in the context of activation functions σ that are hard-limiting, is an equivalent to linear separability and provides a reasonable extension of the notion of linear separability when σ is sigmoidal. Separation of the pre-images of the distinct types of labels, interpreted in terms of the number of sign-crossings,[10] is the key to processor realization in neural networks. Discreteness in $\mathcal{L}_{\underline{w}}(\alpha, \mathcal{P}^n_r(\zeta, \underline{\vartheta}))$ restricts variational consideration in functions over $\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ to sign transitions rather than zero crossings even though σ is defined to be continuous. In the following the number of sign-transitions in the discrete sequences over $\mathcal{L}_{\underline{w}}(\alpha, \mathcal{P}^n_r(\zeta, \underline{\vartheta}))$ will be termed as *order of separability*: the collection of order-0 and order-1 separable sequences (dichotomies) are termed linearly separable. Continuing with the characterization of representation the following result

PROPOSITION 3.2.5 *The number of binary valued functions over the discrete space $\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ that have exactly p sign transitions in the equivalent sequence under any preservance weight in $\wp_n$ is given by*

$$2\binom{|\mathcal{P}^n_r(\zeta, \underline{\vartheta})| - 1}{p} = 2\binom{\frac{2^n(2^{nr}-1)}{2^n-1} - 1}{p},$$

$n = 1, 2, \ldots$; $r = 1, 2, \ldots$; $\zeta \in \Re_+$, $\underline{\vartheta} \in \Re^n$ and $p = 0, 1, \ldots |\mathcal{P}^n_r(\zeta, \underline{\vartheta})| - 1$.

In the above statement, though σ induces only a single sign transition over its domain, multiple sign transitions in the sequences over

[10] In the general case, separation of pre-images is interpreted in terms of level crossings, the level being set to be a mean of the values (or collection of values) that represent the distinct labels

$\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ representing functions over $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ are implied by a lack of monotonicity in the sequence over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ representing points in $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ under a preservance weight $\underline{w} \in \wp_n(\alpha)$, $\alpha \in \Re_+$. (See Figure 3.6 (p. 140) for a clarification of the non-monotonicity in the discrete sequence over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$. All sequences in this figure refer to the input space $\mathcal{P}_1^2(1, \underline{0})$. The example labeled (d) refers to the familiar parity (XOR) problem.) Table 3.2 (p. 144) and Figure 3.7 (p. 145) indicate the number of binary functions over $\mathcal{P}_r^n(1, \underline{0})$ equivalent to binary sequences with exactly p sign transitions expressed as a ratio, denoted by ρ, of the number of distinct binary functions (see the corollary to Proposition 3.2.1 (p. 139)) over the same discrete space for different values of n, r and p:

$$\rho = 2^{-\left(\frac{2^n(2^{nr}-1)}{2^n-1}-1\right)} \binom{\frac{2^n(2^{nr}-1)}{2^n-1} - 1}{p}$$

It is evident from the tabulated values as well as the accompanying graph that the population of (binary) functions with exactly p sign transitions expressed as a ratio of the the number of possible functions shrinks, in general, as the dimensionality n and ranking r are increased for all values of p, $p = 0, 1, \,..|\mathcal{P}_r^n(\zeta, \underline{\vartheta})|$. The reason for this behaviour is simply a consequence of the structure of the function $2^{-k}\binom{k}{p}$, $p = 0, 1, 2, \ldots k$, $k = 1, 2, \ldots$, as indicated in Figure 3.8: values in the isocurves parallel to the coordinate axes show a near binomial distribution. The ratio used above is the mass function (or relative frequency) of function realization with isolated neurons.

Table 3.4: Population of binary functions over $\mathcal{P}_n^n(1,0)$ with order-p separability relative to binary functions over $\mathcal{P}_n^n(1,0)$

r	n	$p=0$	$p=1$	$p=2$	$p=3$	$p=4$	$p=5$	$p=6$	$p=7$
1	1	0 50	0 50	–	–	–	–	–	–
1	2	0 1250	0 3750	0 3750	0 1250	–	–	–	–
1	3	0 0078	0 0547	0 1640	0 2730	0 2730	0 1640	0 0547	0 0078
1	4	0 31 $\times 10^{-4}$	0 0005	0 0032	0 0139	0 0417	0 0916	0 1527	0 1964
1	5	0 47 $\times 10^{-9}$	0 14 $\times 10^{-7}$	0 22 $\times 10^{-6}$	0 21 $\times 10^{-5}$	0 15 $\times 10^{-4}$	0 79 $\times 10^{-4}$	0 0003	0.0012
1	6	0 11 $\times 10^{-18}$	0 68 $\times 10^{-17}$	0 21 $\times 10^{-15}$	0 43 $\times 10^{-14}$	0 65 $\times 10^{-13}$	0 76 $\times 10^{-12}$	0 74 $\times 10^{-11}$	0 60 $\times 10^{-10}$
1	7	0 59 $\times 10^{-38}$	0 75 $\times 10^{-37}$	0 47 $\times 10^{-34}$	0 19 $\times 10^{-32}$	0 61 $\times 10^{-32}$	0 15 $\times 10^{-29}$	0 30 $\times 10^{-28}$	0 53 $\times 10^{-27}$
2	1	0 0313	0 1563	0 3125	0 3125	0 1563	0 0313	–	–
2	2	0 19 $\times 10^{-5}$	0 36 $\times 10^{-4}$	0 0003	0 0018	0.0074	0 0222	0 0518	0 0961
2	3	0 42 $\times 10^{-21}$	0.30 $\times 10^{-19}$	0.10 $\times 10^{-17}$	0 24 $\times 10^{-16}$	0 41 $\times 10^{-15}$	0 55 $\times 10^{-14}$	0 61 $\times 10^{-13}$	0 56 $\times 10^{-12}$
3	1	0 0001	0 0016	0 0095	0 0349	0 0873	0 1571	0 2095	0 2095
3	2	0 10 $\times 10^{-24}$	0 86 $\times 10^{-23}$	0 35 $\times 10^{-21}$	0 95 $\times 10^{-20}$	0 19 $\times 10^{-18}$	0 30 $\times 10^{-17}$	0.39 $\times 10^{-16}$	0 43 $\times 10^{-15}$

r	n	$p = 8$	$p = 9$	$p = 10$
1	1	–	–	–
1	2	–	–	–
1	3	–	–	–
1	4	0 1964	0 1527	0 0916
1	5	0 0037	0 0094	0 0207
1	6	$0\ 42 \times 10^{-9}$	$0\ 26 \times 10^{-8}$	$0\ 14 \times 10^{-7}$
1	7	$0\ 79 \times 10^{-26}$	$0\ 10 \times 10^{-24}$	$0\ 12 \times 10^{-23}$
2	1	–	–	–
2	2	0 1442	0 1762	0 1762
2	3	$0\ 45 \times 10^{-11}$	$0\ 32 \times 10^{-10}$	$0\ 19 \times 10^{-9}$
3	1	0 1571	0 0873	0 0349
3	2	$0\ 41 \times 10^{-14}$	$0\ 34 \times 10^{-13}$	$0\ 25 \times 10^{-12}$

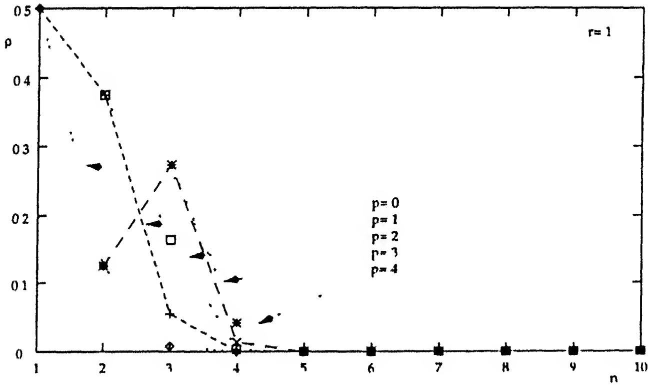


Figure 3.7: Relative population of binary functions over $\mathcal{P}_n^n(1,0)$ with order-p separability

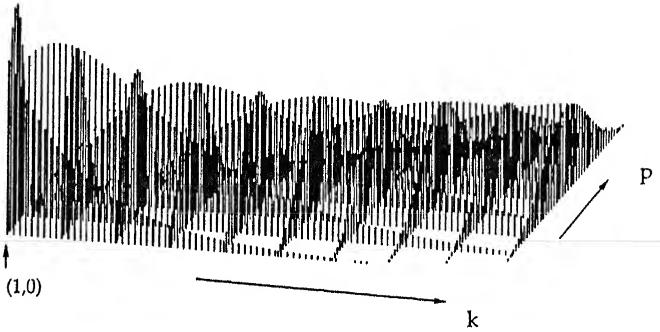


Figure 3.8: Plot of $2^{-k}\binom{k}{p}, p = 0, 1, 2, .. \; k, \; k = 1, 2, ...$

Proposition 3.2.6 *The population of binary functions over $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$, $n = 1, 2, \; ..; r = 1, 2, ...; \zeta \in \Re_+, \underline{\vartheta} \in \Re^n$, with an order of separability no more than p, $p = 0, 1, ...$, in the equivalent sequences obtained under preservance weights drawn from $\wp_n$ decreases as either n, or r or both increase for any given value of p.*

In the above statement, n and r are assumed to be chosen such that the number of sign transitions p does not exceed $|\mathcal{P}_r^n(\zeta, \underline{\vartheta})|$, the number of distinct points in the discrete input space denoted by $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$. As a consequence of the above Proposition even if the activation function is revised to accommodate multiple level-crossings over its domain, as *eg*, (range scaled and translated) Gaussian functions

$$\sigma_g(\xi) = (\zeta_+ - \zeta_-)\, exp(\frac{\xi^2}{2}) + \zeta_-, \;\; \forall \xi \in \Re, \; \zeta_-, \zeta_+ \in \Re, \; \zeta_- < \zeta_+,$$

or (range scaled and translated) Walsh functions

$$\begin{aligned}
\sigma_w^{(0)}(\xi) &= \begin{cases} \zeta_+ & \text{for } \xi \in [-1,1), \\ \zeta_- & \text{for } \xi \in \Re \setminus [-1,1), \end{cases} \\
\sigma_w^{(2\imath+\jmath)}(\xi) &= \sigma_w^{(\imath)}(2\xi-1) + (-1)^{\imath+\jmath}\sigma_w^{(\imath)}(2\xi+1) + 2\zeta_-, \; \forall \xi \in \Re, \\
& \qquad \zeta_-, \zeta_+ \in \Re,\; \zeta_- < \zeta_+,\; \jmath = 0,1,\; i = 0,1, \quad ,
\end{aligned}$$

the proportion of (binary valued) processors realized with such an augmented isolated neuron in relation to the number of processors to be realized always decreases as the number of points in the input space increases (due to the dimensionality n as well as ranking $\imath$). In this thesis, however, σ is assumed to have only one zero crossing and thus only the representation of linearly separable functions will be considered.

PROPOSITION **3.2.7** *The number of binary functions over $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$, $n = 1, 2, ..$; $r = 1, 2, \ldots$; $\zeta \in \Re$ and $\underline{\vartheta} \in \Re^n$ that are linearly separable given any preservance weight in $\wp_n$ is*

$$\frac{2^{n+1}\,(2^{nr}-1)}{(2^n-1)} = 2\,|\mathcal{P}_r^n(\zeta, \underline{\vartheta})|\,.$$

This statement follows from Proposition 3.2.4 (*p*. 141) and Proposition 3.2.5 (*p*. 142). (An idea of the number of linearly separable functions realized by an isolated neuron incorporating any preservance weight is obtained on doubling the entries of Table 3.2 (*p*. 135) for the different values of n and r.) From Proposition 3.2.1 (*p*. 138) the finite length sequences equivalent to (discrete) functions defined over $\mathcal{P}_r^n(\zeta, \underline{\vartheta}) \subset \Re^n$, for given (and admissible) values of n, r, ζ and $\underline{\vartheta}$ are embedded in univariate functions over $\mathcal{L}_{\underline{w}}$, $w \in \wp_n$: it is important to note that the single variable of the functions defined on $\mathcal{L}_{\underline{w}}$ so as to be equivalent to functions over $\Re^n$ is a representation of the closeness, as measured through the inner product operation, between the incident pattern $\underline{x}$ and the preservance weight $\underline{w}$. Functions defined over $\mathcal{L}_{\underline{w}}$ to be representative of (multi-variate) functions defined over the discrete input space $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ exhibit the following feature as a result of the operation of inner product being a continuous mapping.

PROPOSITION **3.2.8** *Under any weight $\underline{w} \in \Re^n$ continuous functions defined over $\Re^n$ and taking values in $\mathcal{Y}$, $\mathcal{Y}$ being a sub-interval of $\Re$, are represented by the operation of inner product as continuous functions over $\mathcal{L}_{\underline{w}}$.*

Continuing with bipolar bivalent activation functions, sign transitions instrumental in the characterization of (binary) functions over $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ through the equivalent sequences defined over the discrete subset of preservation points in $\mathcal{L}_{\underline{w}}$ are possible only through zero crossings. In the following, the zero crossings of functions over $\mathcal{L}_{\underline{w}}$ are assumed to be at points and not through intervals, *ie*, the support of the output value 0 is of null Lesbegue measure. Figure 3.9 (*p.* 149) illustrates the preservation points of $\mathcal{P}_3^2(1, \underline{0})$ in $\mathcal{L}_{\underline{w}}$: the contribution of Boolean hyper-cubes corresponding to the different values of r, $r = 1, 2, 3$, are also indicated. Zero crossings are allowed in the intervals between adjacent preservation points.

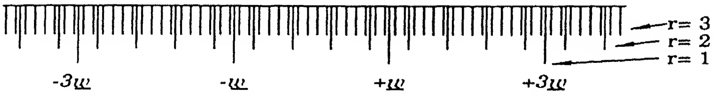


Figure 3 9· Preservation points of $\mathcal{P}_3^2(1,\underline{0})$ in $\mathcal{L}_{\underline{w}}$, $\underline{w} \in \wp_2$

As indicated in the above Figure, the spacing between adjacent preservation points in $\mathcal{L}_{\underline{w}}$ for a preservance weight $\underline{w} \in \wp_n(\alpha)$, $\alpha \in \Re_+$ is not uniform when r is assigned values that are larger than unity. The length of the intervals corresponding to the projection points of $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ in $\mathcal{L}_w$ is either $\zeta 2^{-(nr-1)}$ or one half this value: the latter is applicable only when $r = 2, 3, \ldots$. As the lengths have a common factor, the admissible intervals of zero crossings is $\mathcal{L}_{\underline{w}}$ will be denoted, for $n = 1, 2, \ldots$; $r = 1, 2, \ \ldots$; $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$, by

$$\begin{aligned} {}^{\underline{w}}\Theta_r^n(\alpha,\zeta,\underline{\vartheta})(i) &= \zeta 2^{-nr}\,[\imath - 1, \imath) - \zeta\frac{(2^{nr}-1)}{2^{n(r-1)}} + \underline{w}\,\underline{\vartheta}, \\ & i = 1, 2, \ldots 2(|\mathcal{P}_r^n(\zeta,\underline{\vartheta})| - |\mathcal{P}_{r-1}^n(\zeta,\underline{\vartheta})| - 1), \quad (3.4) \end{aligned}$$

where $\beta_1\mathcal{A} + \beta_2$ is the set $\{\xi = \beta_1\nu + \beta_2 | \nu \in \mathcal{A}\}$ for all $\beta_1, \beta_2 \in \Re$ and $|\mathcal{P}_0^n(\zeta,\underline{\vartheta})| = 0$. The zero crossing intervals ${}^{\underline{w}}\Theta_r^n(\alpha,\zeta,\underline{\vartheta})(i)$ for the values of i indicated in Equation 3.4 are considered as proper subsets of $\left\{\xi = \frac{1}{\|\underline{w}\|}\underline{w}\,\underline{x} \mid \underline{x} \in \mathcal{L}_{\underline{w}}\right\}$, a set isomorphic to $\Re$, however, oriented in the direction of the preservance weight $\underline{w}$.

Proposition 3.2.9 *Functions on $\mathcal{L}_{\underline{w}}$, under a preservance weight $\underline{w} \in \wp_n(\alpha)$, for any $\alpha \in \Re_+$, embed the representation of functions over $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, $n = 1, 2, \ldots$; $r = 1, 2, \ldots$; $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$, which* *have odd number of zero crossings in any interval* ${}^{\underline{w}}\Theta^n_r(\alpha,\zeta,\underline{\vartheta})(\imath)$, $\imath = 1, 2, . \; 2(|\mathcal{P}^n_r(\zeta,\underline{\vartheta})| - |\mathcal{P}^n_{r-1}(\zeta,\underline{\vartheta})| - 1)$, *with a single sign transition between the points in* $\mathcal{L}_{\underline{w}}$ *contained in the boundary of* ${}^{\underline{w}}\Theta^n_r(\alpha,\zeta,\underline{\vartheta})(\imath)$, ie, ${}^{\underline{w}}\overline{\Theta}^n_r(\alpha,\zeta,\underline{\vartheta})(\imath) \setminus {}^{\underline{w}}\Theta^n_r(\alpha,\zeta,\underline{\vartheta})(i)$, *where,* ${}^{\underline{w}}\overline{\Theta}^n_r(\alpha,\zeta,\underline{\vartheta})(i)$ *is the closure of* ${}^{\underline{w}}\Theta^n_r(\alpha,\zeta,\underline{\vartheta})(i)$.

In the ensuing discussion, however, at most one zero crossing is assumed in the intervals between adjacent preservation points of $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ in $\mathcal{L}_{\underline{w}}$. Though the actual number of intervals where zero crossings are admitted is given by $|\mathcal{P}^n_r(\zeta,\underline{\vartheta})| - 1$, Equation 3.4 suggests a larger number while maintaining uniformity in length. A few properties of the zero crossing intervals are listed in the following.

𝒫𝓇𝓸𝓹𝓸𝓈𝓲𝓉𝓲𝓸𝓃 **3.2.10** *Given any preservance weight* $\underline{w} \in \wp_n(\alpha)$, $\alpha \in \Re_+$, *the zero crossing intervals, in* $\mathcal{L}_{\underline{w}}$, *corresponding to functions over* $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, $n = 1, 2, \ldots; r = 1, 2, \ldots; \zeta \in \Re_+$ *and* $\underline{\vartheta} \in \Re^n$, *have the following characteristics with* μ *denoting Lesbegue measure (defined on* $\Re$*) and* $\imath = 1, 2, \ldots 2(|\mathcal{P}^n_r(\zeta,\underline{\vartheta})| - |\mathcal{P}^n_{r-1}(\zeta,\underline{\vartheta})| - 1)$:

1. *Interval shrinkage with dimensionality and ranking:*
   $n_1 > n_2$ *or* $r_1 > r_2$ *implies* $\mu({}^{\underline{w}}\Theta^{n_1}_{r_1}(\alpha,\zeta,\underline{\vartheta})(i)) < \mu({}^{\underline{w}}\Theta^{n_1}_{r_1}(\alpha,\zeta,\underline{\vartheta})(\imath))$, *for all* $n_1, n_2 = 1, 2, \;..$ *and* $r_1, r_2 = 1, 2, \ldots,$.

2. *Interval dilation with scaling:*
   $\mu({}^{\underline{w}}\Theta^n_r(\alpha,\zeta,\underline{\vartheta})(i))$ *increases as* ζ *or* α *increase.*

3 *Independence of interval length to translation:*
$\mu(\underline{w}\Theta^n_r(\alpha, \zeta, \underline{\vartheta})(i))$ *is independent of* $\underline{\vartheta}$ *for all* $\underline{\vartheta} \in \Re^n$

4. *Subdivision of intervals with ranking:*

$$\sum_{i=1}^{2(|\mathcal{P}^n_{r-1}(\zeta,\underline{\vartheta})|-|\mathcal{P}^n_{r-2}(\zeta,\underline{\vartheta})|)} \mu(\underline{w}\Theta^n_{r-1}(\alpha, \zeta, \underline{\vartheta})(i)) = \sum_{j=1}^{2(|\mathcal{P}^n_{r}(\zeta,\underline{\vartheta})|-|\mathcal{P}^n_{r-1}(\zeta,\underline{\vartheta})|)} \mu(\underline{w}\Theta^n_r(\alpha, \zeta, \underline{\vartheta})(j)),$$
$$r = 2, 3, ..$$

Functions over $\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ and $\Re^n$, $n = 1, 2,\ .\ ; r = 1, 2, \ldots; \zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$, have until now been represented in $\mathcal{L}_{\underline{w}}$ using a generic preservance weight $\underline{w} \in \wp_n(\alpha)$, for any $\alpha \in \Re_+$. Recalling the multiplicity of preservance weights for $\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ given an α, as established in § 3.1, the following hold.

PROPOSITION 3.2.11 *For every function* $f \cdot \mathcal{P}^n_r(\zeta, \underline{\vartheta}) \to [\zeta_-, \zeta_+]$, $\mathcal{L}_{\underline{w}} \equiv \mathcal{L}_{-\underline{w}}$ *and* $\mathfrak{L}_{\underline{w}}(\|\underline{w}\|, \mathcal{P}^n_r(\zeta, \underline{\vartheta})) \equiv \mathfrak{L}_{-\underline{w}}(\|\underline{w}\|, \mathcal{P}^n_r(\zeta, \underline{\vartheta}))$ *for all* $\underline{w} \in \Re^n$ *and all admissible values of* n, r, ζ *and* $\underline{\vartheta}$.

This statement is obvious noting that the spaces $\mathcal{L}_{\underline{w}}$ and $\mathfrak{L}_{\underline{w}}$ are defined to capture traversal in the direction of the specified weight $\underline{w}$. Note that in $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{A})$, the notation for the collection of preservation points in $\mathcal{L}_{\underline{w}}$ of a discrete space $\mathcal{A}$ under a preservance weight $\underline{w}$, the suffix $\underline{w}$ denotes the orientation of the collection of preservation points in $\Re^n$ and α denotes the (positive) factor by which the basic vector[11] in the direction of $\underline{w}$ needs to scaled to get the desired weight. For weights $\underline{w} \in \Re^n$ the basic vector in the direction of $\underline{w}$ will be considered to be the same as the unit vector in the direction of $\underline{w}$ and for this reason the scale factor α will be equated to the norm $\|\underline{w}\|$ of the weight $\underline{w}$. It is important to recognize that sequences defined on $\mathfrak{L}_{\underline{w}}(\|\underline{w}\|, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ relate to the sequences on $\mathfrak{L}_{-\underline{w}}(\|\underline{w}\|, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ in the following manner.

Proposition 3.2.12 *For every function $f\colon \mathcal{P}_r^n(\zeta, \underline{\vartheta}) \rightarrow [\zeta_-, \zeta_+]$ realized by an isolated neuron with a weight $\underline{w} \in \Re^n$, the function realized by the weight $-\underline{w}$, denoted by $f_-\colon \mathcal{P}_r^n(\zeta, \underline{\vartheta}) \rightarrow [\zeta_-, \zeta_+]$, is a* complement *of f in the sense that*

$$\forall \underline{x} \in \mathcal{P}_r^n(\zeta, \underline{\vartheta}) \quad f(\underline{x}) + f_-(\underline{x}) = \zeta_+ + \zeta_-.$$

Proposition 3.2.13 *For every function $f\colon \mathcal{P}_r^n(\zeta, \underline{\vartheta}) \rightarrow [\zeta_-, \zeta_+]$ there exist preservance weights $\underline{w}_1, \underline{w}_2 \in \wp_n(\alpha)$, $\underline{w}_1 \neq \underline{w}_2$ and $\underline{w}_2 \neq -\underline{w}_1$, such that for any $\alpha \in \Re_+$, the sequences over $\mathfrak{L}_{\underline{w}_1}(\alpha, \mathcal{P}_r^n(\alpha, \underline{\vartheta}))$ and $\mathfrak{L}_{\underline{w}_2}(\alpha, \mathcal{P}_r^n(\alpha, \underline{\vartheta}))$ representing f are identical when expressed in terms of a traversal over the spaces $\mathfrak{L}_{\underline{w}_1}(\alpha, \mathcal{P}_r^n(\alpha, \underline{\vartheta}))$ and $\mathfrak{L}_{\underline{w}_2}(\alpha, \mathcal{P}_r^n(\alpha, \underline{\vartheta}))$, respectively, rather than over the domain space $\mathcal{P}_r^n(\alpha, \underline{\vartheta})$ of f.*

Proof: The statement will be established through the following example with the accompanying illustration.

[11] For the preservance weights considered in the preceding discussion, the basic vector is one of the $2^n\, n!$ weights provided by the assignment suggested in the proof of Theorem 3.1.1 (p. 113).

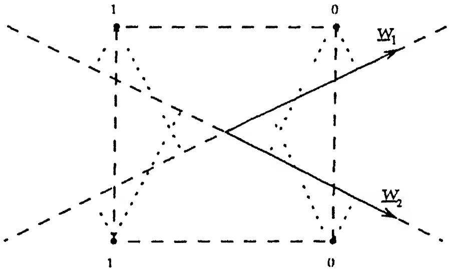


Figure 3.10· Equivalence between preservance weights

For the discrete input set $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, with the associated hypotheses on n, r, ζ and $\underline{\vartheta}$, consider any preservance weight $\underline{w}_1$ from the collection $\wp_n(\alpha)$, for any appropriate value of α, $\alpha \in \Re_+$. Consider now the weight $\underline{w}_2$ constructed from $\underline{w}_1$ as $w_{2_i} = w_{1_i}$, $i = 1,2,\quad n$, $i \neq j$ and $w_{2_j} = -w_{1_j}$ for some $j = 1,2,\quad n$ or as $w_{2_i} = w_{1_i}$, $i = 1,2,.\ n$, $i \neq j_1$, $i \neq j_2$ and $w_{2_{j_1}} = w_{1_{j_2}}$, $w_{1_{j_2}} = w_{2_{j_1}}$, for some $j_1, j_2 = 1,2,\ldots n$. It is obvious that $\underline{w}_1 \in \wp_n(\alpha)$ implies $\underline{w}_2 \in \wp_n(\alpha)$. If we now consider sequences $s_1\cdot \mathcal{L}_{\underline{w}_1}(\alpha,\mathcal{P}_r^n(\zeta,\underline{\vartheta})) \to \mathcal{B}$ and s_2: $\mathcal{L}_{\underline{w}_2}(\alpha,\mathcal{P}_r^n(\zeta,\underline{\vartheta})) \to \mathcal{B}$ such that $\forall \underline{x} \in \mathcal{P}_r^n(\zeta,\underline{\vartheta}) \quad f(\underline{x}) = s_{1_i} = s_{2_j}\quad i = \underline{w}_1\ \underline{x}, j = \underline{w}_2\cdot\underline{x}$, then we find that

$$\forall \underline{x}_1, \underline{x}_2 \in \mathcal{P}_r^n(\zeta,\underline{\vartheta}) \quad (s_{1_{i_1}}, s_{1_{i_2}}) \in \boldsymbol{R} \equiv (s_{2_{j_1}}, s_{2_{j_2}}) \in \boldsymbol{R}$$

for any (ordering) relation $\boldsymbol{R} \subseteq \mathcal{B} \times \mathcal{B}$, where $i_1 = \underline{w}_1\ \underline{x}_1$, $i_2 = \underline{w}_1\ \underline{x}_2$, $j_1 = \underline{w}_2\cdot\underline{x}_1$ and $j_2 = \underline{w}_2\ \underline{x}_2$. A typical example for the relation $\boldsymbol{R}$ is the 'less than or equal to' relation denoted by $\leq$.

□

An immediate implication of the above statement is that the number of linearly separable functions realizable by an isolated neuron is not given simply as the number of distinct preservance weights (as given by Proposition 3.1.2 (*p* 119)) multiplied by the number of linearly separable functions realized through any preservance weight (see Proposition 3.2.7 (*p*. 147)), but is to be sought out through attempts at understanding the (algebraic) structure of preservance weights. However, an investigation into the algebraic properties of the class of preservance weights is beyond the scope of this thesis.

3.3 Learning of Preservance Weights and Generalization in Isolated Neurons

Functions realized by isolated neurons are decided by the weights $\underline{w}$ and threshold θ, as discussed in Chapter 2 It is imperative that the weights and threshold are automatically specified given, information in terms of inputs, with corresponding (possibly partially specified) outputs, related to the required mapping: this automatic specification has been termed learning. The collection of inputs, drawn from $\mathcal{P}_r^n(\zeta, \underline{\vartheta}) \subset \Re^n$ for a suitable choice of r, ζ and $\underline{\vartheta}$, together with the corresponding requirement on the assignments to the output constitutes the training set. For convenience, the collection of inputs contained in a training set will be denoted by $\mathcal{T}_\imath$, $\mathcal{T}_i \subseteq \mathcal{P}_r^n(\zeta, \underline{\vartheta})$, and the projection of these inputs along a weight $\underline{w} \in \wp_n(\alpha)$, for any $\alpha \in \Re_+$, will be denoted by $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{T}_i)$.

A key issue in function realization through isolated neurons in addition to learning is that of a specification of assignments to the output corresponding to inputs not contained in the training set, *ie* generalization: this issue too influences the choice of weights and threshold. Maintaining the tradition of neural network research, this discussion will consider only the problem of specifying weights and thresholds in the scope of learning and generalization and will not dwell on the issue of specifying the nature of activation function σ

In view of the preservation of $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ in $\mathcal{L}_{\underline{w}}$, specification of a function over $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ is equivalent to the specification of a univariate function over $\mathcal{L}_{\underline{w}}$ at the finitely many distinct points given by $\mathfrak{L}_{\underline{w}}(\alpha,\mathcal{P}^n_r(\zeta,\underline{\vartheta}))$. Thus, generalization viewed as a problem of incorporating the functional characteristics specified in the training set to an input space equaling the entirety of either $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ or $\Re^n$ reduces to a problem of function extension – commonly addressed in discussions of functional analysis – and under the preservance weights, generalization amounts to extending a function defined typically over a subset of $\mathfrak{L}_{\underline{w}}(\alpha,\mathcal{P}^n_r(\zeta,\underline{\vartheta})) \subset \mathcal{L}_{\underline{w}}$, for any $\alpha \in \Re_+$, to the entirety of $\mathcal{L}_{\underline{w}}$.

I will begin by considering the case wherein all inputs in $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ are considered, together with the corresponding outputs, in the training set, *ie*, $\mathcal{T}_\imath = \mathcal{P}^n_r(\zeta,\underline{\vartheta})$. Generalization is restricted in this case to a function extension from $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ to $\Re^n$, equivalently the discrete set of preservation points $\mathfrak{L}_{\underline{w}}(\alpha,\mathcal{P}^n_r(\zeta,\underline{\vartheta})) \subset \mathcal{L}_{\underline{w}}$ to the entirety of $\mathcal{L}_{\underline{w}}$, where

$\underline{w} \in \wp_n(\alpha)$ for any $\alpha \in \Re_+$. For simplicity of reasoning, the activation function σ is assumed to be of the hard-limiting type (see § 2.2). Under this assumption, functions that are defined over $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ are essentially two-valued and the following definition is invoked for reasons of clarity in the discussion

𝔇𝔈𝔉𝔍𝔑𝔍𝔗𝔍𝔒𝔑 **3.3.1** *A function* $\mathfrak{D}$ $\mathcal{P}_r^n(\zeta, \underline{\vartheta}) \rightarrow \{\zeta_-, \zeta_+\}$, $\zeta_-, \zeta_+ \in \Re$, $\zeta_- \neq \zeta_+$, *is termed a* dichotomy.[12] *If* $\zeta_- = -\zeta_+$, *the dichotomy is termed bipolar. Dichotomies that are onto for the entire range space* $\{\zeta_-, \zeta_+\}$ *are termed* non-trivial. *In the context of processor representation with (isolated) neurons, the mapping is considered surjective over a subset* $\mathcal{T}_i$ *of* $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$. *A dichotomy is termed* complete *if the mapping is surjective on the entirety of* $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$.

In an isolated neuron weights $\underline{w}$ that preserve $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ in $\mathcal{L}_{\underline{w}}$ belong to $\wp_n$: the cardinality of the class of preservance weights is more than unity for all values of n. Though all weights in $\wp_n$ preserve the discrete space $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ in $\mathcal{L}_{\underline{w}}$, the linear subspace (of $\Re^n$) is different for weights $\underline{w}$ which differ in orientation. In the following, the dependence of function representation on the choice of weights is formally stated.

𝔓ℜ𝔒𝔓𝔒𝔖𝔍𝔗𝔍𝔒𝔑 **3.3.1** *The sequences over* $\mathcal{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ *representing functions (including dichotomies) over* $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ *under a preservance weight* $\underline{w}$ *vary as* $\underline{w}$ *varies over* $\wp_n$.

[12] The notion of a dichotomy has already been made use of in Chapter 2 while detailing the available characterization of isolated neurons and networks of such neurons.

This aspect has been dealt in considerable detail in § 3.2. However, the following is to be noted in the context of selecting weights in isolated neurons: the first of these statements is equivalent to item 2 in Proposition 3.2.10 (p. 150).

$\mathcal{PROPOSITION}$ **3.3.2** *Variation in preservance weights due to scaling, as decided by α, alters the sequences over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ (representing functions over $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$) only in terms of scale and not in terms of sign transitions.*

$\mathcal{PROPOSITION}$ **3.3.3** *Variation of weights $\underline{w}$ over $\wp_n(\alpha)$ for any $\alpha \in \Re_+$ influence the number and location of sign transitions in the sequences over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ representing functions over $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$.*

Invariance of the number and location of sign transitions (*ie*, zero crossings) in the sequences to specific variation of preservance weights, as indicated in Proposition 3.2.13 (p. 152), should, however, be noted. Recalling the consequence of linear separability on the nature of sequences admissible on $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$, as indicated by Proposition 3.2.4 (p. 141), the problem of learning of weights, corresponding to a linear separable dichotomy, in an isolated neuron is equivalent to one of finding a weight $\underline{w}$ in $\wp_n$ such that the number of sign transitions in the sequences over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ representing bipolar bipolar dichotomies, over $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ is no more that unity, thereby leading to the following.

Proposition **3.3.4** *A bipolar dichotomy* $\mathfrak{D}$: $\mathcal{P}_r^n(\zeta,\underline{\vartheta}) \to \{\zeta_-,\zeta_+\}$ *is linearly separable if there exists a weight* $\underline{w} \in \wp_n(\alpha)$, *for any* $\alpha \in \Re_+$, *such that no more than one sign transition occurs in the bipolar bivalent sequence over* $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta,\underline{\vartheta}))$ *representing* $\mathfrak{D}$

Theorem **3.3.1** *Bipolar dichotomies exhibiting an invariance in the number of sign transitions for all preservance weights in* $\wp_n(\alpha)$ *for all* $\alpha \in \Re_+$ *are either constant (and, hence, trivially linearly separable) or not linearly separable,* $n = 2, 3, \ldots$. *There are only two linearly separable bipolar dichotomies* $\mathfrak{D}$: $\mathcal{P}_r^n(\zeta,\underline{\vartheta}) \to \{\zeta_-,\zeta_+\}$ *independent of the dimensionality* n, viz, *functions which assign uniformly one of* ζ_- *and* ζ_+ *to all points in* $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$.

Proof: Invariance in the number of sign transitions to preservance weights $\underline{w} \in \wp_n(\alpha)$ for any $\alpha \in \Re_+$ of a constant function is obvious. The only bipolar bivalent functions that exhibit this feature are those that assign, uniformly, one of ±1 to all points in $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$.

Functions that represent dichotomies that are linearly separable cannot also exhibit invariance in the number of sign transitions to all preservance weights in $\wp_n(\alpha)$ as established in the following. Given that a weight $\underline{w}_1 \in \wp_n(\alpha)$ represents a linearly separable dichotomy, consider, if possible, the same dichotomy to be represented by a weight $\underline{w}_2$ derived from $\underline{w}_1$ as below:

$$w_{2_j} = -w_{1_j}, \text{ for some } j = 1,2,\ldots n,\ w_{2_i} = w_{1_i},\ i = 1,2,\ \ldots n,\ i \neq j.$$

As j is varied, for at least one value of j the sequence over the discrete space $\mathfrak{L}_{\underline{w}_2}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ will have at least two sign transitions. This leads to a contradiction and, thereby, the necessary statement is established.

The argument invoked above also shows that the only functions that exhibit invariance in the number of sign transitions are those that have an assignment that is invariant to permutations in the weights. It is now simple to see that such functions are, indeed, not linearly separable, a typical example being the parity function (XOR when the input space dimensionality $n = 2$)

□

(The construction used in the proof of the above statement has already been used in the proof of Proposition 3.2.13 (*p.* 152).)

Proposition **3.3.5** *For every non-trivial dichotomy $\mathfrak{D}\cdot$ $\mathcal{P}_r^n(\zeta, \underline{\vartheta}) \to \{\zeta_-, \zeta_+\}$, $n = 1, 2, \ldots$; $r = 1, 2, \ldots$; $\zeta \in \Re_+$, $\underline{\vartheta} \in \Re^n$ and $\zeta_-, \zeta_+ \in \Re^n$, $\zeta_- < \zeta_+$, there exists at least two weights in $\wp_n(\alpha)$, for any $\alpha \in \Re_+$, such that the bipolar bivalent sequence over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ has more than one sign transition.*

Proof: Refer the proofs of Proposition 3.2.13 (*p.* 152) and Theorem 3.3.1 (*p.* 158). Given any non-trivial dichotomy $\mathfrak{D}$, there exists a weight $\underline{w} \in \wp_n(\alpha)$ such that the sequence over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ has at least two sign transitions: this is obvious in the event the given dichotomy is not linearly separable and Theorem 3.3.1 (p. 158) assures the existence of such a weight in case of linearly separable dichotomies. The argument of Proposition 3.2.13 (p. 152) is applicable to all bipolar bivalent functions over $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ and assures the existence of the second weight given a knowledge of a weight satisfying the stated hypothesis. Note that the collection of preservance weights $\wp_n(\alpha)$, for any $\alpha \in \Re_+$, is a complementary structure in the sense that $\forall \underline{w} \in \wp_n(\alpha)$, $\underline{w} \in \wp_n(\alpha) \Rightarrow -\underline{w} \in \wp_n(\alpha)$. Thus multiple sign transitions in the sequences over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta,\underline{\vartheta}))$, corresponding to non-trivial bipolar bivalent functions over $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, are caused by at least four distinct preservance weights (but two distinct orientations).

□

The above Proposition implies that an isolated neuron cannot realize all dichotomies on $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$. (Such a result has long been known in relation to dichotomies on $\mathcal{B}^n$.) In addition, the above Proposition shows that only a subclass of $\wp_n$ need be searched for a solution to the learning problem of linear separable dichotomies on $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$: the identification of this subclass is based on the number of sign transitions (zero crossings) in the sequences over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta,\underline{\vartheta}))$, where $\underline{w}$ refers to a candidate weight in $\wp_n$. The following definition is introduced to provide a criterion to aid the selection of preservance weights.

Definition 3.3.2 *Given a function f: $\mathcal{P}_r^n(\zeta,\underline{\vartheta}) \rightarrow [\zeta_-, \zeta_+]$ (dichotomy $\mathfrak{D}$: $\mathcal{P}_r^n(\zeta,\underline{\vartheta}) \rightarrow \{\zeta_-, \zeta_+\}$), a preservance weight $\underline{w} \in \wp_n(\alpha)$, for any $\alpha \in$*

$\Re_+$, *ıs termed* admissible for $f\ (\mathfrak{D})$ *if, in the sequence over* $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ *representing* $f\ (\mathfrak{D})$ *over* $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$*, the number of sign transıtions does not exceed that accommodated by the activation functıon* σ*.*

Proposition 3.3.4 (*p.* 158) provides an interpretation of linear separability in terms of partitions on $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ effected by sign transitions (zero crossings). (In the same vein, the general situation of order-p separability can be interpreted in terms of partitions induced by level crossings. However, for reasons of convenience, only bipolar dichotomies that are also linearly separable – *ie*, order-0 and order-1 – are considered in this thesis.) The criterion of admissibility of weights restricts choice of preservance weights to those that ensure a realization of the desired dichotomy. From the preceding dıscussion it is easy to establish the following.

THEOREM **3.3.2** *The problem of learning a given non-trıvial complete bipolar dichotomy* $\mathfrak{D}\cdot\ \mathcal{P}_r^n(\zeta, \underline{\vartheta}) \rightarrow \{\zeta_-, \zeta_+\}$ *in an isolated neuron with a hard-limıting activation functıon involves the two distinct steps:*

1. *Enumerate the weights* $\underline{w} \in \wp_n(\alpha)$*, for any suıtable* $\alpha \in \Re_+$ *tıll either* $\underline{w}$ *is admissible for* $\mathfrak{D}$ *(*ie*, the number of sıgn transitions in the sequence over* $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ *representing the gıven dichotomy does not exceed unity) or the space of preservance weıghts is exhausted. In the latter case the dichotomy is not linearly separable.*

2. *In the event a preservance weight admissible for $\mathfrak{D}$ is obtained* (ie, *the given dichotomy is lınearly separable) assign a value to the threshold θ from the set ${}^{\underline{w}}\Theta_{\imath}^{n}(\alpha, \zeta, \underline{\vartheta})(i)$, where $\imath$, the index of sign transition location is identıfied by the relatıon*·[13]

$$\mathfrak{D}\Big|_{(\underline{w}\ \mathcal{P}_r^n(\zeta,\underline{\vartheta}))\cap \bigcup_{\jmath=1}^{\imath-1} {}^{\underline{w}}\overline{\Theta}_r^n(\alpha,\zeta,\underline{\vartheta})(\jmath)}$$
$$\equiv -\mathfrak{D}\Big|_{(\underline{w}\ \mathcal{P}_r^n(\zeta,\underline{\vartheta}))\cap \bigcup_{\jmath=\imath+1}^{2(|\mathcal{P}_r^n(\zeta,\underline{\vartheta})|-|\mathcal{P}_{r-1}^n(\zeta,\underline{\vartheta})|-1)} {}^{\underline{w}}\overline{\Theta}_r^n(\alpha,\zeta,\underline{\vartheta})(\jmath)}$$

where, $\imath = 1, 2, \ . \ 2(|\mathcal{P}_r^n(\zeta, \underline{\vartheta})| - |\mathcal{P}_{r-1}^n(\zeta, \underline{\vartheta})| - 1)$ *and* $\underline{w}\ \ \mathcal{A}$ *denotes the collectıon of inner products, wıth $\underline{w}$, of the elements of $\mathcal{A}$, for any $\mathcal{A} \subset \Re^n$; $\underline{w}\ \ \mathcal{A} \subset \Re$.*

The above theorem shows that learning involves an enumerative procedure for weights (see Figure 3.2 (*p.* 122)) and a search for the threshold in a linearly ordered space and, thereby, provides the basis for an algorithm for learning in isolated neurons: the details of such an algorithm will not, however, be taken up in this discussion. Any element of the interval[14] ${}^{\underline{w}}\Theta_r^n(\zeta, \underline{\vartheta})(i)$, where i is determined as indicated in step 2 of the above theorem, is allowed to be a candidate for the threshold. No commitment is, however, made on the nature of assignment to $\mathfrak{D}$ over the interval ${}^{\underline{w}}\Theta_r^n(\zeta, \underline{\vartheta})(\imath)$, $i = 1, 2, .\ . 2(|\mathcal{P}_r^n(\zeta, \underline{\vartheta})| - |\mathcal{P}_{r-1}^n(\zeta, \underline{\vartheta})| - 1)$, which contains the threshold θ. As a consequence of Proposition 3.2.13 (*p.* 152) the following is important to note.

[13] Note that $\zeta_- = -\zeta_+$ in a bıpolar dichotomy.

[14] As the interval is defined to be left closed, the function represented will be right continuous.

PROPOSITION **3.3.6** *The preservance weights $\underline{w} \in \wp_n(\alpha)$, for any $\alpha \in \Re_+$, admissible to a given bipolar dichotomy $\mathfrak{D}$ are not unique.*

In order that the admissibility of a preservance weight selected through the learning procedure is not altered, generalization, viewed as a situation of function extension from $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$, for any $\alpha \in \Re_+$, $n = 1, 2, .\ ; r = 1, 2, ..\ ; \zeta \in \Re_+$, $\underline{\vartheta} \in \Re^n$, to $\mathcal{L}_{\underline{w}}$, is expected to preserve the number and location, at least to the extent of the interval ${}^{\underline{w}}\Theta_r^n(\alpha, \zeta, \underline{\vartheta})(\imath)$, $\imath = 1, 2,\ .\ 2(|\mathcal{P}_r^n(\zeta, \underline{\vartheta})| - |\mathcal{P}_{r-1}^n(\zeta, \underline{\vartheta})| - 1)$, of sign transitions (*ie*, zero crossings) in the assignments made to points in $\mathcal{L}_{\underline{w}}$ given the assignments to points in the discrete space $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$. Note that in view of Proposition 3.3.3 (*p*. 157), the index of the interval in which the threshold belongs is not altered by α, $\alpha \in \Re_+$, though, the specific nature of interval is very much influenced by α.

Having discussed the case wherein the training set consists of a complete dichotomy, I will now consider the more general and realistic case wherein the training set does not contain all points of $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ in the training set, *ie*, $\mathcal{T}_\imath \subset \mathcal{P}_r^n(\zeta, \underline{\vartheta})$, the containment being proper. As a consequence the collection of inputs contained in the training set when projected along any preservance weight $\underline{w} \in \wp_n(\alpha)$, for any $\alpha \in \Re_+$, will form a proper subset (*viz*, $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{T}_\imath)$) of the discrete set $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta})) \subset \mathcal{L}_{\underline{w}}$. The weight and threshold obtained in the learning of a dichotomy on $\mathcal{T}_\imath$ are expected to be related to those obtained in the learning of a dichotomy on $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$.

Generalization, in this case, involves two components: one of extending functions defined on $\mathcal{T}_\imath$ to $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$, equivalently extending functions defined on $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{T}_\imath)$ to $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$, and the other is of extending functions defined on $\mathcal{T}_\imath$ to $\Re^n \setminus \mathcal{P}_r^n(\zeta, \underline{\vartheta})$, equivalently extending functions defined on $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{T}_\imath)$ to $\mathcal{L}_{\underline{w}} \setminus \mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$. Here again, the operational criterion of generalization is a preservation of the number, and location, of sign transitions in the function over $\mathcal{L}_{\underline{w}}$ given the assignment over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{T}_\imath)$, $\underline{w} \in \wp_n(\alpha)$ for any $\alpha \in \Re_+$ Additionally, the process of generalization is also expected to preserve the number and location of sign transitions in the sequence over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ given the assignment over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{T}_\imath)$

Proposition 3.3.7 *Learning a bipolar dichotomy wherein* $\mathcal{T}_\imath \subset \mathcal{P}_r^n(\zeta, \underline{\vartheta})$ *for admissible values of* n, $\imath$, ζ *and* $\underline{\vartheta}$, *is equivalent to the problem of learning a complete bipolar dichotomy once the assignments to points in* $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta})) \setminus \mathfrak{L}_{\underline{w}}(\alpha, \mathcal{T}_\imath)$, $\underline{w} \in \wp_n(\alpha)$ *for any* $\alpha \in \Re_+$, *is completed by the process of generalization.*

Proposition 3.3.8 *Given a bipolar dichotomy* $\tilde{\mathfrak{D}}: \mathcal{T}_\imath \to \{\zeta_-, \zeta_+\}$, $\mathcal{T}_\imath \subset \mathcal{P}_r^n(\zeta, \underline{\vartheta})$, $n = 1, 2, ..$; $r = 1, 2, \ldots$; $\zeta \in \Re_+$ *and* $\underline{\vartheta} \in \Re^n$, *every preservance weight* $\underline{w} \in \wp_n(\alpha)$, *for any* $\alpha \in \Re_+$, *admissible for the bipolar dichotomy* $\mathfrak{D}\ \mathcal{P}_r^n(\zeta, \underline{\vartheta}) \to \{\zeta_-, \zeta_+\}$ *is also admissible for* $\tilde{\mathfrak{D}}$, *wherein* $\mathfrak{D}|_{\mathcal{T}_\imath} \equiv \tilde{\mathfrak{D}}$.

The above Proposition suggests that the preservance weights admissible in the representation of a dichotomy $\mathfrak{D}$ are inherited as admissible preservance weights of the dichotomy $\tilde{\mathfrak{D}}$: with an interpretation of $\mathfrak{D}$ and $\tilde{\mathfrak{D}}$ as subsets of the Cartesian product $\mathcal{P}_r^n(\zeta, \underline{\vartheta}) \times \{\zeta_-, \zeta_+\}$ it is easy to see that $\tilde{\mathfrak{D}} \subseteq \mathfrak{D}$. Generalization, in view of the above inheritance of admissible preservance weights, is to play the role of ensuring that given a dichotomy $\tilde{\mathfrak{D}}$, the dichotomy $\mathfrak{D}$ suggested as an extension of $\tilde{\mathfrak{D}}$ will have, as admissible preservance weight(s), at least one of the several preservance weights admissible to $\tilde{\mathfrak{D}}$. The converse of Proposition 3.3.8 holds when the completion $\mathfrak{D}$ of a given dichotomy $\tilde{\mathfrak{D}}$ ensures that the number of zero crossings in the sequence over the collection of projection points $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$, $\alpha \in \Re_+$, representing the desired function on $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ is no different from that in the sequence over $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{T}_i)$ representing the given function on $\mathcal{T}_i$.

Learning, as an approach of specifying an admissible preservance weight $\underline{w}$ given a partially specified function and identification of a region, in $\mathcal{L}_{\underline{w}}$, within which the threshold, θ, is located, is not restricted to the case of dichotomies alone and is easily extended to the more general situation of representing processors mapping $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ to $[\zeta_-, \zeta_+] \subset \Re$ as established in the following: the discussion is restricted to the case of isolated neurons with sigmoidal activation function for reasons of convenience. In this case too, the considerations of generalization are identical to the situation wherein hard-limiting activation functions are used. Processors are assumed, in the following, to realize bipolar functions: this assumption allows learning to be characterized on the lines of Theorem 3.3.2 (*p*. 161).

$\mathcal{THEOREM}$ **3.3.3** *Representation of processors $f\cdot \mathcal{T}_\imath \rightarrow [\zeta_-, \zeta_+]$, $\mathcal{T}_i \subseteq \mathcal{P}^n_r(\zeta, \underline{\vartheta})$, in an isolated neuron with a sigmoidal activation function involves the two distinct steps:*

1. *Preservance weight enumeration – this is identical to step 1 ın* Theorem 3.3.2 (*p.* 161), *except that the enumeration seeks for a preservance weight admıssible to f.*

2. *Threshold range ıdentification – this is similar to step 2 in* Theorem 3.3.2 (*p.* 161). *θ, the threshold, is assıgned a value from the set $^{\underline{w}}\Theta^n_r(\zeta, \underline{\vartheta})(\imath)$, the index i, $\imath = 1, 2, \ldots 2(|\mathcal{P}^n_r(\zeta, \underline{\vartheta})|\,|\mathcal{P}^n_{r-1}(\zeta, \underline{\vartheta})| - 1)$, being given by*

$$sgn(f)\,|_{(\underline{w}\ \mathcal{P}^n_r(\zeta,\underline{\vartheta}))\cap \bigcup_{\jmath=1}^{\imath-1} {}^{\underline{w}}\overline{\Theta}^n_r(\alpha,\zeta,\underline{\vartheta})(\jmath)}$$

$$\equiv -sgn(f)\,|_{(\underline{w}\ \mathcal{P}^n_r(\zeta,\underline{\vartheta}))\cap \bigcup_{\jmath=\imath+1}^{2(|\mathcal{P}^n_r(\zeta,\underline{\vartheta})|-|\mathcal{P}^n_{r-1}(\zeta,\underline{\vartheta})|-1)} {}^{\underline{w}}\overline{\Theta}^n_r(\alpha,\zeta,\underline{\vartheta})(\jmath)}$$

where, $sgn(\cdot)$ is the sign function

$$\forall \xi \in \Re \quad sgn(\xi) = \begin{cases} -1 & \textit{if}\ \xi < 0, \\ 0 & \textit{if}\ \xi = 0, \\ +1 & \textit{otherwise.} \end{cases}$$

A point to note in Theorem 3.3.2 (*p.* 161) and Theorem 3.3.3 above, in particular, the steps wherein the interval of sign transition (in $\mathcal{L}_{\underline{w}}$) is identified, is that the index i, $\imath = 1, 2, \ldots 2(|\mathcal{P}^n_r(\zeta, \underline{\vartheta})|\,|\mathcal{P}^n_r(\zeta, \underline{\vartheta})| - 1)$, of the set $^{\underline{w}}\Theta^n_r(\alpha, \zeta, \underline{\vartheta})(i)$ is, in general, not unique.

Table 3.5 (*p*. 168) and Table 3.6 (*p*. 169) provide examples of functions that are to be realized through an isolated neuron with a function extension: these examples show associations between $\mathcal{P}_3^2(1,\underline{0})$ and $\{-1,+1\}$. In the dichotomies indicated, members of the training set $\mathcal{T}_\imath$ are emboldened. Normal entries refer to the association resulting from a generalization (function extension). An enumeration (see Figure 3.2 (*p*. 122) for the scheme) of the preservance weights of $\mathcal{P}_3^2(1,\underline{0})$ is provided in Table 3 7 (*p*. 170). The ensuing discussion easily extends to the case when the sign of the neuron outputs, rather than the outputs themselves are expected to be in the set $\{-1,+1\}$. (However, in the latter case, it is important to recognize that the specific values realized depend on the type of activation function σ.)

In an isolated neuron, a function on the discrete input space $\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ under a preservance weight $\underline{w}$ is equivalent to a sequence on the collection of projection points $\mathfrak{L}_{\underline{w}}(\alpha,\mathcal{P}_r^n(\zeta,\underline{\vartheta}))$. Table 3.8 (*p*. 170) indicates the sequences that would result due to functions f_1 and f_2 indicated in Table 3.5 (*p*. 168) and Table 3.6 (*p*. 169). Sequences corresponding to weights $\underline{w}_{<0>}$ and $\underline{w}_{<1>}$ only are considered noting the equivalence indicated in Proposition 3.2 13 (*p*. 152). $s_\imath^\epsilon$ in Table 3.8 denotes the sequence over $\frac{1}{\|\underline{w}_{<\epsilon>}\|}\underline{w}_{<\epsilon>} \cdot \mathfrak{L}_{\underline{w}_{<\epsilon>}}(\alpha, P_r^n(\zeta,\underline{\vartheta}))$ representing the function $f_\imath$, $i = 1, 2$, $\epsilon = 0, 1$. Table 3.8 indicates that the preservance weight $\underline{w}_{<0>}$ is not admissible for function f_1 while $\underline{w}_{<1>}$ is admissible. In the case of function f_2 no preservance weight in $\wp_n$ is admissible, *ie*, f_2 is not a linearly separable dichotomy.

Table 3.5: Association $f_1 \, . \, \mathcal{P}_3^2(1, \underline{0}) \to \{-1, +1\}$

(-1 3125,-1 3125)	↦	1	(-1 3125,-1 1875)	↦	1	(-1 25,-1 25)	↦	1
(-1.1875,-1 3125)	↦	1	(-1 1875,-1 1875)	↦	1	(-1 3125,-0 8125)	↦	1
(-1 3125,-0 6875)	↦	1	(-1 25,-0 75)	↦	1	(-1 1875,-0 8125)	↦	1
(-1 1875,-0 6875)	↦	1	(-1,-1)	↦	1	(-0 8125,-1 3125)	↦	1
(-0 8125,-1 1875)	↦	1	**(-0.75,-1.25)**	↦	**1**	(-0 6875,-1 3125)	↦	1
(-0 6875,-1 1875)	↦	1	(-0 8125,-0 8125)	↦	1	(-0 8125,-0 6875)	↦	1
(-0 75,-0 75)	↦	1	(-0 6875,-0 8125)	↦	1	(-0 6875,-0 6875)	↦	1
(-1 3125,0 6875)	↦	1	(-1 3125,0 8125)	↦	1	**(-1.25,0.75)**	↦	**1**
(-1 1875,0 6875)	↦	1	(-1 1875,0 8125)	↦	1	(-1 3125,1 1875)	↦	1
(-1 3125,1 3125)	↦	1	(-1 25,1 25)	↦	1	(-1 1875,1 1875)	↦	1
(-1 1875,1 3125)	↦	1	**(-1,1)**	↦	**1**	(-0 8125,0 6875)	↦	1
(-0 8125,0 8125)	↦	1	(-0 75,0 75)	↦	1	(-0 6875,0 6875)	↦	1
(-0 6875,0 8125)	↦	1	(-0 8125,1 1875)	↦	1	(-0 8125,1 3125)	↦	1
(-0 75,1.25)	↦	1	(-0 6875,1 1875)	↦	1	(-0 6875,1 3125)	↦	1
(0 6875,-1 3125)	↦	-1	(0 6875,-1 1875)	↦	-1	(0 75,-1 25)	↦	-1
(0 8125,-1 3125)	↦	-1	(0 8125,-1 1875)	↦	-1	(0 6875,-0 8125)	↦	-1
(0 6875,-0 6875)	↦	-1	(0 75,-0 75)	↦	-1	(0 8125,-0 8125)	↦	-1
(0 8125,-0 6875)	↦	-1	(1,-1)	↦	-1	**(1.1875,-1.3125)**	↦	**-1**
(1 1875,-1 1875)	↦	-1	(1 25,-1 25)	↦	-1	(1 3125,-1 3125)	↦	-1
(1 3125,-1 1875)	↦	-1	(1 1875,-0 8125)	↦	-1	(1 1875,-0 6875)	↦	-1
(1.25,-0.75)	↦	**-1**	(1 3125,-0 8125)	↦	-1	(1 3125,-0 6875)	↦	-1
(0 6875,0 6875)	↦	1	(0 6875,0 8125)	↦	1	(0 75,0 75)	↦	1
(0 8125,0 6875)	↦	1	(0 8125,0 8125)	↦	1	(0 6875,1.1875)	↦	1
(0 6875,1 3125)	↦	1	(0 75,1 25)	↦	1	(0 8125,1 1875)	↦	1
(0 8125,1 3125)	↦	1	**(1,1)**	↦	**1**	(1 1875,0 6875)	↦	1
(1 1875,0.8125)	↦	1	**(1.25,0.75)**	↦	**1**	(1 3125,0 6875)	↦	1
(1 3125,0 8125)	↦	1	(1 1875,1 1875)	↦	1	(1 1875,1.3125)	↦	1
(1 25,1 25)	↦	1	(1 3125,1.1875)	↦	1	(1 3125,1 3125)	↦	1

Table 3.6: Association $f_2 \cdot \mathcal{P}_3^2(1,\underline{0}) \to \{-1,+1\}$

(-1 3125,-1 3125)	↦ -1	(-1 3125,-1 1875)	↦ -1	(-1 25,-1 25)	↦ -1
(-1 1875,-1 3125)	↦ -1	(-1 1875,-1 1875)	↦ -1	(-1 3125,-0 8125)	↦ -1
(-1 3125,-0 6875)	↦ -1	(-1 25,-0 75)	↦ -1	(-1 1875,-0 8125)	↦ -1
(-1 1875,-0 6875)	↦ -1	(-1,-1)	↦ -1	(-0 8125,-1 3125)	↦ -1
(-0 8125,-1 1875)	↦ -1	**(-0.75,-1.25)**	**↦ -1**	(-0 6875,-1 3125)	↦ -1
(-0 6875,-1 1875)	↦ -1	(-0 8125,-0 8125)	↦ -1	(-0 8125,-0 6875)	↦ -1
(-0 75,-0 75)	↦ -1	(-0 6875,-0 8125)	↦ -1	(-0 6875,-0 6875)	↦ -1
(-1 3125,0 6875)	↦ 1	(-1 3125,0 8125)	↦ 1	**(-1.25,0.75)**	**↦ 1**
(-1 1875,0 6875)	↦ 1	(-1 1875,0 8125)	↦ 1	(-1 3125,1 1875)	↦ 1
(-1 3125,1 3125)	↦ 1	(-1 25,1 25)	↦ 1	(-1 1875,1 1875)	↦ 1
(-1 1875,1 3125)	↦ 1	**(-1,1)**	↦ 1	(-0 8125,0 6875)	↦ 1
(-0 8125,0 8125)	↦ 1	(-0 75,0 75)	↦ 1	(-0 6875,0 6875)	↦ 1
(-0 6875,0 8125)	↦ 1	(-0 8125,1 1875)	↦ 1	(-0 8125,1 3125)	↦ 1
(-0 75,1 25)	↦ 1	(-0 6875,1 1875)	↦ 1	(-0 6875,1 3125)	↦ 1
(0 6875,-1 3125)	↦ 1	(0 6875,-1.1875)	↦ 1	(0 75,-1 25)	↦ 1
(0 8125,-1 3125)	↦ 1	(0 8125,-1 1875)	↦ 1	(0 6875,-0 8125)	↦ 1
(0 6875,-0 6875)	↦ 1	(0 75,-0 75)	↦ 1	(0 8125,-0 8125)	↦ 1
(0 8125,-0 6875)	↦ 1	(1,-1)	↦ 1	**(1.1875,-1.3125)**	↦ 1
(1 1875,-1 1875)	↦ 1	(1 25,-1 25)	↦ 1	(1 3125,-1 3125)	↦ 1
(1 3125,-1 1875)	↦ 1	(1 1875,-0 8125)	↦ 1	(1 1875,-0 6875)	↦ 1
(1.25,-0.75)	↦ 1	(1 3125,-0 8125)	↦ 1	(1 3125,-0 6875)	↦ 1
(0 6875,0 6875)	↦ -1	(0 6875,0 8125)	↦ -1	(0 75,0 75)	↦ -1
(0 8125,0.6875)	↦ -1	(0 8125,0.8125)	↦ -1	(0.6875,1 1875)	↦ -1
(0 6875,1 3125)	↦ -1	(0 75,1 25)	↦ -1	(0 8125,1 1875)	↦ -1
(0 8125,1 3125)	↦ -1	**(1,1)**	↦ -1	(1 1875,0 6875)	↦ -1
(1 1875,0.8125)	↦ -1	**(1.25,0.75)**	↦ -1	(1 3125,0.6875)	↦ -1
(1 3125,0 8125)	↦ -1	(1 1875,1.1875)	↦ -1	(1 1875,1 3125)	↦ -1
(1 25,1 25)	↦ -1	(1 3125,1 1875)	↦ -1	(1 3125,1 3125)	↦ -1

Table 3.7: Preservance weights of $\mathcal{P}_3^2(1, \underline{0})$ with $\alpha = 1$

$\underline{w}_{<0>} = [+1\ +2]$	$\underline{w}_{<1>} = [-1\ +2]$
$\underline{w}_{<2>} = [+1\ -2]$	$\underline{w}_{<3>} = [-1\ -2]$
$\underline{w}_{<4>} = [+2\ +1]$	$\underline{w}_{<5>} = [-2\ +1]$
$\underline{w}_{<6>} = [+2\ -1]$	$\underline{w}_{<7>} = [-2\ -1]$

Table 3.8: Sequences representing functions f_1 and f_2

	s_1^0	s_1^1	s_2^0	s_2^1		s_1^0	s_1^1	s_2^0	s_2^1		s_1^0	s_1^1	s_2^0	s_2^1
3 9375	1	1	1	1	3 8125	1	1	-1	1	3 75	1	1	1	1
-3 6875	1	1	-1	1	3 5625	1	1	1	1	3 4375	1	1	-1	1
-3 3125	1	1	1	1	3 25	1	1	1	1	3 1875	1	1	1	1
3 0625	1	1	-1	1	3	1	1	1	1	2 9375	1	1	1	1
2 8125	1	1	1	1	2 75	1	1	1	1	2 6875	1	1	1	1
-2 5625	1	1	1	1	2 4375	1	1	1	1	2 3125	1	1	1	1
2 25	1	1	1	1	2 1875	1	1	1	1	2 0625	1	1	1	1
1 9375	1	1	1	1	1 8125	1	1	1	1	1 75	1	1	1	1
-1 6875	1	1	1	1	1 5625	1	1	1	1	-1 4375	1	1	1	1
-1 3125	1	1	1	1	-1 25	1	1	1	1	-1 1875	1	1	1	1
1 0625	1	1	1	1	1	1	1	1	1	0 9375	1	1	1	1
0 8125	1	1	1	1	0 75	1	1	1	-1	0 6875	1	1	1	1
0 5625	1	1	1	1	0 4375	1	1	1	1	0 3125	1	1	1	1
0 25	1	1	1	1	0 1875	1	1	1	1	0 0625	1	1	1	1
0 0625	1	1	1	1	0 1875	1	1	1	1	0 25	1	1	1	1
0 3125	1	1	1	1	0 4375	1	1	1	1	0 5625	1	1	1	1
0 6875	1	1	1	1	0 75	-1	1	1	1	0 8125	1	1	1	-1
0 9375	1	1	1	1	1	1	1	1	1	1 0625	1	1	1	1
1 1875	1	1	1	1	1 25	1	1	1	1	1 3125	1	1	1	1
1 4375	1	1	1	1	1 5625	1	1	1	-1	1 6875	1	1	1	1
1 75	1	1	1	1	1 8125	1	1	1	1	1 9375	1	1	1	1
2 0625	1	-1	1	1	2 1875	1	-1	1	1	2 25	1	-1	1	1
2 3125	1	-1	1	1	2 4375	1	1	1	1	2 5625	1	1	1	1
2 6875	1	-1	1	1	2 75	1	1	1	1	2 8125	1	1	-1	1
2 9375	1	-1	1	1	3	1	1	1	1	3 0625	1	-1	1	1
3 1875	1	-1	1	1	3 25	1	1	1	1	3 3125	1	1	1	1
3 4375	1	-1	1	1	3 5625	1	1	1	1	3 6875	1	1	1	1
3 75	1	-1	-1	1	3 8125	1	-1	1	1	3.9375	1	-1	-1	1

Unlike the hitherto accepted notion of learning by examples, the approach to learning, in the sense of an automatic specification of weight and threshold values given a dichotomy, even though discussed at the level of isolated neurons, requires only a single instance of (valid) assignment to each of the inputs included in the training set. The procedure of specifying weight and threshold values rather than being iterative is enumerative in nature and is quite attractive in situations where on-line learning is necessary.

3.4 Preservation in Higher Radix Input Spaces

Preservation of input spaces and the associated simplification in the learning procedure have been discussed in the foregoing in the specific case wherein the elements of the discrete space $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$, for admissible values of r, ζ and $\underline{\vartheta}$, correspond to a binary number system.[15] As the notion of preservance is based on the operational correspondence between inner product and positional numbering systems, it is readily apparent that the notion of preservation of discrete input spaces by appropriately chosen weights is not restricted to collections of binary vectors, or vector collections derived from binary spaces. The following observation of the preservation in numbering systems with a radix r is analogous to Theorem 3.1.1 (*p*. 113).

[15] Note that $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ is constructed from scaled and translated versions of the space of binary inputs $\mathcal{B}^n$

THEOREM **3.4.1** *A weight vector $\underline{w}$ given by $w_i = \alpha\mathfrak{r}^{i-1}$, $i = 1, 2, .\ ., n$, for any $\alpha \in \Re_+$, preserves in $\mathcal{L}_{\underline{w}}$ all the $\mathfrak{r}^n$ points of $\{0, 1, .\ \ , \mathfrak{r}-1\}^n$, a discrete space of radix $\mathfrak{r}$, $\mathfrak{r} = 2, 3, .\ .$ for all $n = 1, 2,\ \ .$*

In the following I will consider the following discrete spaces of radix $\mathfrak{r}$, $\mathfrak{r} = 2, 3, \ldots$, the elements being real numbers.

$$
\begin{aligned}
{}_{\mathfrak{r}}\mathcal{H} &= \begin{cases} \left\{-\frac{\mathfrak{r}-1}{2},\ ..-\frac{3}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, .\ ., \frac{\mathfrak{r}-1}{2}\right\}, & \text{if } \mathfrak{r} \text{ is even,} \\ \left\{-\frac{\mathfrak{r}-1}{2},\ .\ \ -2, -1, 0, 1, 2,\ .., \frac{\mathfrak{r}-1}{2}\right\}, & \text{otherwise,} \end{cases} && (3.5) \\
{}_{\mathfrak{r}}\mathcal{H}^n(\zeta, \underline{\vartheta}) &\triangleq \zeta\,{}_{\mathfrak{r}}\mathcal{H}^n + \underline{\vartheta}, && (3.6)
\end{aligned}
$$

where, $n = 1, 2, ..\ ;\ \zeta \in \Re_+$, $\vartheta \in \Re^n$ and ${}_{\mathfrak{r}}\mathcal{H}^n$ is the n-fold Cartesian product of ${}_{\mathfrak{r}}\mathcal{H}$. (${}_{\mathfrak{r}}\mathcal{H}(\zeta, \underline{\vartheta}) \equiv {}_{\mathfrak{r}}\mathcal{H}^1(\zeta, \underline{\vartheta})\ \forall\zeta \in \Re_+$, $\underline{\vartheta} \in \Re^n$.) The collection of scaled and translated binary vectors introduced in § 3.1 are specific instances of the discrete sets indicated by Equation 3.6: $\mathcal{B}^n(\zeta, \underline{\vartheta}) \equiv {}_2\mathcal{H}^n(2\zeta, \underline{\vartheta})$ for all $n = 1, 2, .\ .;\ \zeta \in \Re_+$, and $\vartheta \in \Re^n$. A discrete space ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ (analogous to $\mathcal{P}^n_r(\zeta, \underline{\vartheta})$), $\mathfrak{r} = 2, 3,\ \ \ $, is also considered through the following recursive construction

$$
\begin{aligned}
{}_{\mathfrak{r}}\mathcal{P}^n_0(\zeta, \underline{\vartheta}) &= \emptyset, && (3\ 7\text{a}) \\
{}_{\mathfrak{r}}\mathcal{P}^n_1(\zeta, \underline{\vartheta}) &= {}_{\mathfrak{r}}\mathcal{H}^n(\zeta, \underline{\vartheta}), && (3.7\text{b}) \\
{}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta, \underline{\vartheta}) &= {}_{\mathfrak{r}}\mathcal{P}^n_{r-1}(\zeta, \underline{\vartheta}) \bigcup \left(\bigcup_{i=1}^{\mathfrak{r}^{nr}} {}_{\mathfrak{r}}\mathcal{H}^n(\mathfrak{r}^{-(r-1)n}\zeta, \underline{\vartheta}^{(r)}_i + \underline{\vartheta}) \right), \\
& \qquad r = 2, 3,\ .., && (3.7\text{c})
\end{aligned}
$$

where, $\underline{\vartheta}^{(r)}_i$, $i = 1, 2, ..\ \mathfrak{r}^{nr}$, are the $\mathfrak{r}^{nr}$ (ordered) points of the discrete set ${}_{\mathfrak{r}}\mathcal{P}^n_{r-1}(\zeta, \underline{\vartheta}) \setminus {}_{\mathfrak{r}}\mathcal{P}^n_{r-2}(\zeta, \underline{\vartheta})$, $n = 1, 2, \ldots;\ \zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$.

The centroid of ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, for all admissible values of $\mathfrak{r}$, n, r, ζ and $\underline{\vartheta}$, is the same as that of ${}_{\mathfrak{r}}\mathcal{H}^n(\zeta,\underline{\vartheta})$ in the above construction. It is also important to note that in the above construction of ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, the centroidal element of the scaled and shifted discrete space ${}_{\mathfrak{r}}\mathcal{H}^n(\zeta,\underline{\vartheta})$ is made to coincide with an appropriate element of ${}_{\mathfrak{r}}\mathcal{P}^n_i(\zeta,\underline{\vartheta})$, $i = 1,2,\ldots, r-1$. As a consequence, the structure of ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, is not the same for even and odd radices $\mathfrak{r}$ except for the case when r, the ranking index, is unity. ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ with an odd radix $\mathfrak{r}$ will, in general, have several points of different ranks coinciding as compared to the case when $\mathfrak{r}$ is even The following is a consequence of the above construction.

Proposition 3.4.1 *The number of distinct points of $\Re^n$ included in the discrete space ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, $\mathfrak{r} = 2,3,\ldots; n = 1,2,\ldots; r = 1,2,\ldots; \zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$, is given by*

$$|{}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})| = \begin{cases} \dfrac{\mathfrak{r}^n(\mathfrak{r}^{nr}-1)}{\mathfrak{r}^n - 1}, & \textit{if } \mathfrak{r} \textit{ is even,} \\ \mathfrak{r}^{nr}, & \textit{otherwise.} \end{cases}$$

The growth in cardinality of ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ as a function of the ranking index r, $r = 1,2,\ldots$, is similar in nature to that demonstrated in $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, $n = 1,2,\ldots$, $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$, including the property of denseness in $\mathcal{L}_{\underline{w}}$ for a preservance weight $\underline{w} \in \wp_n(\alpha)$, for any $\alpha \in \Re_+$: hence, these aspects of ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ are not being established explicitly. In a similar way the interval subdivisions in $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}^n_r(\zeta,\underline{\vartheta}))$, for any $\alpha \in \Re^n$ and admissible values for n, r, ζ and ϑ, are also seen in the discrete space $\mathfrak{L}_{\underline{w}}(\alpha, {}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta}))$ corresponding to ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ for all radices $\mathfrak{r}$,

$\mathfrak{r} = 2, 3, \ldots$. A few other properties of the discrete spaces ${}_{\mathfrak{r}}\mathcal{P}^{n}_{r}(\zeta, \underline{\vartheta})$ are stated in the following.

PROPOSITION 3.4.2 *For distinct radices $\mathfrak{r}_1$ and $\mathfrak{r}_2$, $\mathfrak{r}_1, \mathfrak{r}_2 = 2, 3, \;.,$ such that $\mathfrak{r}_1$ and $\mathfrak{r}_2$ are both even (or both odd) and $\mathfrak{r}_1 < \mathfrak{r}_2$, ${}_{\mathfrak{r}_1}\mathcal{P}^{n}_{r}(\zeta, \underline{\vartheta}) \subset {}_{\mathfrak{r}_2}\mathcal{P}^{n}_{r}(\zeta, \underline{\vartheta})$ for all admissible values of n, r, ζ and $\underline{\vartheta}$.*

PROOF: Noting that ${}_{\mathfrak{r}_1}\mathcal{H} \subset {}_{\mathfrak{r}_2}\mathcal{H}$ for the hypothesis considered on $\mathfrak{r}_1$ and $\mathfrak{r}_2$, the statement is an immediate consequence of the construction of the discrete space ${}_{\mathfrak{r}}\mathcal{P}^{n}_{r}(\zeta, \underline{\vartheta})$.

□

COROLLARY TO PROPOSITION 3.4.2 ${}_{\mathfrak{r}_1}\mathcal{P}^{n}_{r_1}(\zeta, \underline{\vartheta}) \subset {}_{\mathfrak{r}_2}\mathcal{P}^{n}_{r_2}(\zeta, \underline{\vartheta})$ *for all $\mathfrak{r}_1, \mathfrak{r}_2$ such that $\mathfrak{r}_1$ and $\mathfrak{r}_2$ are both even, or both odd, $\mathfrak{r}_1 < \mathfrak{r}_2$, $\mathfrak{r}_1, \mathfrak{r}_2 = 1, 2, .\;.;$ $r_1 < r_2$, $r_1, r_2 = 2, 3, \ldots$ and admissible values of n, ζ and $\underline{\vartheta}$*

PROOF: From Equation 3.7 (*p.* 172) it is clear that ${}_{\mathfrak{r}}\mathcal{P}^{n}_{r_1}(\zeta, \underline{\vartheta}) \subset {}_{\mathfrak{r}}\mathcal{P}^{n}_{r_2}(\zeta, \underline{\vartheta})$ for all admissible values of $\mathfrak{r}$, n, ζ and $\underline{\vartheta}$ when $r_1 < r_2$, $r_1, r_2 = 1, 2, \ldots$. The necessary statement is an immediate consequence of this observation combined with Proposition 3.4.2.

□

In view of Theorem 3.4.1 (*p.* 172), the preservation of ${}_{\mathfrak{r}}\mathcal{P}^{n}_{r}(\zeta, \underline{\vartheta})$ is assured as in the following.

$\mathcal{T}\mathcal{H}\mathcal{E}\mathcal{O}\mathcal{R}\mathcal{E}\mathcal{M}$ **3.4.2** *Weights $\underline{w}$ given by the assignment*

$$w_i \in \bigcup_{j=1}^{n} \mathcal{B}(\alpha\mathfrak{r}^{j-1}, 0),\ \alpha \in \Re_+,\ i = 1, 2, .\ \ n, \tag{3.8}$$

subject to the restriction that $|w_i| \neq |w_k|$ for all $i, k = 1, 2, .\ \ n$, $i \neq k$, preserve, in $\mathcal{L}_{\underline{w}}$, all points of the discrete space ${}_{\mathfrak{r}}\mathcal{P}_r^n(\zeta, \underline{\vartheta})$, $\mathfrak{r} = 2, 3, \ldots$; $n = 1, 2,\ \ $; $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$.

The proof of this statement follows exactly on the lines of Theorem 3.1.2 (*p.* 115). In the following, the collection of preservance weights of the discrete space ${}_{\mathfrak{r}}\mathcal{P}_r^n(\zeta, \underline{\vartheta})$, for the admissible values of $\mathfrak{r}$, n, r, ζ and $\underline{\vartheta}$, will be denoted by ${}_{\mathfrak{r}}\wp_n$ and the restriction of preservance weights to any specific value of $\alpha \in \Re^n$ is denoted by ${}_{\mathfrak{r}}\wp_n(\alpha)$. (Note that this notation is analogous to $\wp_n$ and $\wp_n(\alpha)$, respectively; $\wp_n \equiv {}_2\wp_n$, $\wp_n(\alpha) \equiv {}_2\wp_n(\alpha)$.) The following characteristic of the collection of preservance weights is an immediate consequence of Theorem 3.4.2 (*p.* 175).

$\mathcal{P}\mathcal{R}\mathcal{O}\mathcal{P}\mathcal{O}\mathcal{S}\mathcal{I}\mathcal{T}\mathcal{I}\mathcal{O}\mathcal{N}$ **3.4.3** *For all radices $\mathfrak{r}$, $\mathfrak{r} = 2, 3, .\ .$,*

$$|{}_2\wp_n(\alpha)| = |{}_3\wp_n(\alpha)|\ \ .\ \ = |{}_{\mathfrak{r}}\wp_n(\alpha)| \ldots = n!\, 2^n,$$

for any $\alpha \in \Re^n$ and all $n = 1, 2, \ldots$.

Consider a number representation system whose radix, denoted by $\mathfrak{r}_r$, is given recursively by

$$\mathfrak{r}_1 = 2, \tag{3.9}$$

$$\mathfrak{r}_r = 2^n(\mathfrak{r}_{r-1} - 1) + 3,\ r = 2, 3,\ \ .. \tag{3.10}$$

Then the following characterization of preservance using weights in $\wp_n$ relating ranking and radices is interesting.

𝒯ℋℰ𝒪ℛℰℳ **3.4.3** *$\mathcal{P}_r^n(\zeta, \underline{\vartheta})$, $r = 1, 2, .\;$, is the largest subset of the discrete subspace ${}_{\mathfrak{r}_r}\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ that is preserved, in $\mathcal{L}_{\underline{w}}$, by weights $\underline{w} \in \wp_n(\alpha)$ for any $\alpha \in \Re_+$ and admissible values of n, ζ and $\underline{\vartheta}$.*

𝒫ℛ𝒪𝒪ℱ: In view of the construction of the spaces $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ and ${}_{\mathfrak{r}}\mathcal{P}_r^n(\zeta, \underline{\vartheta})$, it is immediately apparent that the smallest radix $\mathfrak{r}$ for which $\mathcal{P}_r^n(\zeta, \underline{\vartheta}) \subset {}_{\mathfrak{r}}\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ is given by Equation 3.10. Noting that every weight $\underline{w} \in \wp_n(\alpha)$, for any $\alpha \in \Re_+$, preserves all points of $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ in $\mathcal{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ through the discrete subset $\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$, a study of the influence of points in ${}_{\mathfrak{r}}\mathcal{P}_r^n(\zeta, \underline{\vartheta}) \setminus \mathcal{P}_r^n(\zeta, \underline{\vartheta})$ under weights $\underline{w} \in \wp_n(\alpha)$ shows that there exists at least one pair of points, say $(\underline{x}_1, \underline{x}_2)$, such that $\underline{x}_1 \in \mathcal{P}_r^n(\zeta, \underline{\vartheta})$ and $\underline{x}_2 \in {}_{\mathfrak{r}}\mathcal{P}_r^n(\zeta, \underline{\vartheta}) \setminus \mathcal{P}_r^n(\zeta, \underline{\vartheta})$ and $\underline{w} \cdot \underline{x}_1 = \underline{w}\, \underline{x}_2$ for the specific weight $\underline{w} \in \wp_n(\alpha)$. The associated breakdown of the one-one correspondence disallows $\underline{w} \in \wp_n(\alpha)$ from being a preservance weight for discrete spaces having components from ${}_{\mathfrak{r}}\mathcal{P}_r^n(\zeta, \underline{\vartheta}) \setminus \mathcal{P}_r^n(\zeta, \underline{\vartheta})$ as well as $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$. It is also important to recognize that since the points in ${}_{\mathfrak{r}}\mathcal{P}_r^n(\zeta, \underline{\vartheta}) \setminus \mathcal{P}_r^n(\zeta, \underline{\vartheta})$ are, in general, not uniquely mapped into $\mathcal{L}_{\underline{w}}$ under inner product involving a weight $\underline{w} \in \wp_n(\alpha)$, the elements of $\wp_n(\alpha)$ are disallowed from being preservance weights for discrete spaces that are subsets of ${}_{\mathfrak{r}}\mathcal{P}_r^n(\zeta, \underline{\vartheta}) \setminus \mathcal{P}_r^n(\zeta, \underline{\vartheta})$.

□

From the foregoing, it is important to note that though the collections of preservance weights are equinumerous for a given dimensionality n and ranking r, as stated in Proposition 3.4.3 (p. 175), the distinctness in the preservance weights as established in Theorem 3.4.3 (p. 176) prevents isomorphisms from being established between the preservance weights corresponding to the discrete spaces ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ for different radices $\mathfrak{r}$, $\mathfrak{r} = 1, 2, \ldots$ Moreover, given an $\alpha \in \Re_+$, $\|\underline{w}\|$ increases with the radix $\mathfrak{r}$ for all weights $\underline{w} \in \wp_n(\alpha)$ and, hence, as the radix $\mathfrak{r}$ increases vectors derived from the preservance weights $\underline{w}$ as $\frac{\underline{w}}{\|\underline{w}\|}$ tend to cluster around the n (unit norm) basis vectors of $\Re^n$. (Note that $\wp_n(\alpha) \subset \Re^n$ for all $\alpha_+ \in \Re_+$.) This bunching of preservance weights corresponding to the discrete spaces ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$, as the radix $\mathfrak{r}$ increases greatly diminishes the utility, from the point of view of preservance, of enlargements of the subsets of $\Re^n$ preserved, in $\mathcal{L}_{\underline{w}}$, under inner product with preservance weight $\underline{w}$.

In the foregoing, the analysis has been one of finding preservance weights given a discrete subset of $\Re^n$, and the discrete subsets have been chosen to be derived from the basic space B^n by means of scaling and translation. Despite the limitation of weight bunching with an increase in the radix of representation, preservance weights and the associated preservance, are not restricted to a trivial discrete subset of $\Re^n$ as indicated in the following.

THEOREM 3.4.4 *Given any weight $\underline{w} \in \Re^n$, $\|\underline{w}\| \neq 0$, there exists a discrete subset of $\Re^n$, in one-one correspondence with $\mathcal{B}^n$, all of whose points are preserved in $\mathcal{L}_{\underline{w}}$.*

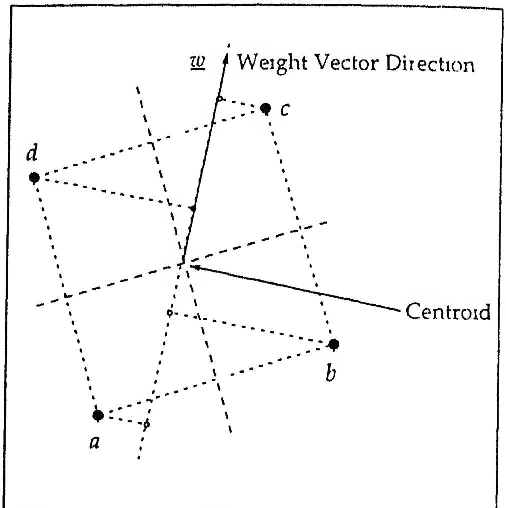


Figure 3.11: Preservance input space $\frac{\underline{w}}{2}\mathcal{H}^n(1,0)$

PROOF: A constructive proof is provided.

Consider the discrete subset of $\Re^n$ given by the collection of points $\underline{x}$, $\|\underline{x}\| = \sqrt{n}$, such that

$$\underline{w}\,\underline{x} = \imath\zeta, i = -(2^n - 1). \quad , -3, -1, 1, 3, \ ..(2^n - 1),$$

for some appropriate value of $\zeta \in \Re_+$. An illustration of this space for $n = 2$ and $\mathfrak{r} = 2$ accompanies in Figure 3.11 (*p.* 178): the discrete space is made up of points of $\Re^n$ labeled a, b, c and d.

One-one correspondence of the discrete space constructed above with the discrete space $\mathcal{B}^n$ is obvious from the construction.

□

In the following, I will denote n dimensional discrete spaces constructed as in the above with a radix $\mathfrak{r}$ numbering by $\frac{\underline{w}}{\mathfrak{r}}\mathcal{H}^n(\zeta, \underline{\vartheta})$, $\zeta \in \Re_+$

being the scale factor, $\underline{\vartheta} \in \Re^n$ the translation and the prescript $\underline{w}$ indicating explicitly the vector $\underline{w}$ with respect to which the space is constructed. On the lines of ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ a discrete space denoted by ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ will be constructed recursively as in the following.

$$
\begin{aligned}
{}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_0(\zeta,\underline{\vartheta}) &= \emptyset, & \text{(3.11a)}\\
{}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_1(\zeta,\underline{\vartheta}) &= {}^{\underline{w}}_{\mathfrak{r}}\mathcal{H}^n(\zeta,\underline{\vartheta}), & \text{(3.11b)}\\
{}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta}) &= {}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_{r-1}(\zeta,\underline{\vartheta}) \bigcup \left(\bigcup_{\imath=1}^{\mathfrak{r}^{nr}} {}^{\underline{w}}_{\mathfrak{r}}\mathcal{H}^n(\mathfrak{r}^{-(r-1)n}\zeta, \underline{\vartheta}_\imath^{(r)} + \underline{\vartheta}) \right), &\\
&\qquad r = 2,3,\ldots, & \text{(3.11c)}
\end{aligned}
$$

where, $\underline{\vartheta}_\imath^{(r)}$, $\imath = 1,2,\ldots \mathfrak{r}^{nr}$, are the $\mathfrak{r}^{nr}$ (ordered) points of the discrete set ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_{r-1}(\zeta,\underline{\vartheta}) \setminus {}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_{r-2}(\zeta,\underline{\vartheta})$, $n = 1,2,\ldots$; $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$. It is immediately apparent that the discrete spaces ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{H}^n(\zeta,\underline{\vartheta})$ and ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ incorporate rotation of the basic space $\mathcal{B}^n$ in addition to scaling and translation. The spaces ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{H}^n_r(\zeta,\underline{\vartheta})$ and ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ being constructed such that the given weight vector $\underline{w} \in \Re^n$, $\|\underline{w}\| \neq 0$, preserves all points of these spaces in $\mathcal{L}_{\underline{w}}$, these spaces will be termed *preservance input spaces* corresponding to the weight $\underline{w}$. For the sake of completeness, it is worthwhile to note that ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta}) \equiv {}^{\underline{w}_{<0>}}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ for all admissible values of $\mathfrak{r}$, n, ζ and $\underline{\vartheta}$; $\underline{w}_{<0>}$, as indicated earlier, being the weight $\underline{w}$ whose elements are given by $w_\imath = \mathfrak{r}^{i-1}$, $i = 1,2,\ldots n$.

As suggested in Theorem 3.1.2 (*p*. 115), the weight vector $\underline{w}$, $\underline{w} \in \Re^n$, $\|\underline{w}\| \neq 0$, will not be the lone preservance weight for the preservance input space ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta}) \subset \Re^n$, and the collection of preservance weights for ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ will be denoted by ${}^{\underline{w}}_{\mathfrak{r}}\wp_n$. Definitionally, the weight vector

$\underline{w}$ is an element of ${}^{\underline{w}}_{\mathfrak{t}}\wp_n$. Proposition 3.4.3 (*p.* 175) immediately implies that $|{}^{\underline{w}}_{\mathfrak{t}}\wp_n|$ is $n!\,2^n$, the same as $|{}_{\mathfrak{t}}\wp_n(\alpha)|$, for any $\alpha \in \Re_+$. Note that the collection of preservance weights ${}^{\underline{w}}_{\mathfrak{t}}\wp_n$ for the preservance input space ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ corresponding to a given $\underline{w}$ is in one-one correspondence with ${}_{\mathfrak{t}}\wp_n(\alpha)$ for all $\alpha \in \Re_+$. In addition, the discrete spaces $\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ and ${}^{\underline{w}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ are also in one-one correspondence for all admissible values of n, r, ζ and $\underline{\vartheta}$. This one-one correspondence allows an extension of the representational characteristics discussed earlier in the context of the discrete input space $\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ to function realization situations involving the discrete set ${}^{\underline{w}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ as the domain. However, the details of such an extension are beyond the scope of this thesis. In § 4.3, I will consider the issue of identification of preservance input spaces given a collection on input vectors. Such an identification relates to the problems of learning and generalization in neural networks.

3.5 Summary

Isolated neurons, the functional basis in the connectionist (*ie*, neural network) approach to information processing, have been studied from the point of view of the representation potential for signal processors. Preservance of discrete space, assured for all non-null weights admissible in an isolated neuron, restricts this study to functions defined on discrete spaces: the discrete input space preserved is, in general, a subset of a lattice points embedded in the Euclidean space $\Re^n$ of dimensionality n. As preservation of the discrete input spaces is not restricted by the direction of the weight vector, input space dimensionality and radix of numbering (radices higher than 2 are considered only in the input space, the output space always being binary) this notion is useful in simplifying functions of several discrete variables to equivalent sequences on one-dimensional (discrete) spaces.

The interplay between the issues of learning and generalization in isolated neurons has been studied through the simplification of function representation enabled by preservance of discrete spaces: while learning addresses the issue of specifying values for weights and threshold given examples of association between inputs and outputs, generalization concerns the equally important issue of extending the function to the region of the input space not covered by the training set. Generalization is not without a criterion: the criterion, specified through either a test set (different from the training set) or qualitative specifications, inevitably, amounts to a formulation in terms of the number of sign transitions (zero-crossings) in the equivalent sequence of the function being represented relative to the number of sign transitions (zero-crossings) that are accommodated by the activation function.

Processor representation in isolated neurons, as discussed in this chapter, is limited to processors (functions) that on discrete spaces preserved by a weight, say $\underline{w} \in \Re^n$ demand an assignment which when projected as the equivalent sequence along the one-dimensional space described by $\underline{w}$ have no more sign-transitions (zero crossings) than that accommodated by the activation function: in the case of sigmoidal (including hard-limiting) activation functions, this restricted collection of processors is termed linearly separable. The limited sense in which processor representation is provided by isolated neurons necessitates a study of the representational characteristics of networked ensembles of neurons. Preservance being restricted to discrete spaces, it is of interest to know the possibility of representing symbolic computation through neural networks: this interest stems partially from the existing result, established by **Lippmann** (1987) and others, that with at least two layers of neural processing all Boolean functions of several variables can be represented.

Chapter 4

Layered Neural Signal Processing

[I]t is worth pondering the fact . that a universal computer could be built entirely out of linear threshold modules. This does not in any sense reduce the theory of computation and programming to the theory of perceptrons. Some philosophers might like to express the relevant general principle by saying that the computer is so much more than the sum of its parts that the computer scientist can afford to ignore the nature of the components and consider only their connectivity.

— Marvin Minsky and Seymour Papert
in *Perceptrons: An ıntroduction to computatıonal geometry*,
MIT Press, Cambridge, MA, 1990

Isolated neurons incorporating hard-limiting activation functions are capable of representing interesting discrete functions, however, suffer from the serious limitation of being able to represent, as indicated in Chapter 3, a shrinking fraction of the total number of possible functions. This limitation is seen in neurons equipped with sigmoidal activation functions too, the limitation being seen as a lack of denseness of the space of functions represented by isolated neurons measured relative to the space of continuous functions.

Neuronal ensembles, investigated in the literature with a view to overcome the limitations in representation of isolated neurons, have principally been of the layered variety. Most common types are structures which incorporate feed-forward connections and/or lateral interaction; the details of these structures have already been presented in Chapter 2. (An alternate approach has been that of incorporating polynomial, or higher order, discriminants in the isolated neuron. In this approach the effect of non-linear association across layers in multilayered networks is sought to be provided, equivalently, by higher-order interactions between the elements of the input pattern.)

In this chapter, a logical continuation of the discussion initiated in the previous chapter, I will take up a study of layered neural networks. Neural signal processors,[1] as these processing structures are termed,

[1] Apart from the issue that the neural processing paradigm allows for learning (of internal representations) through examples, the main difference between the neural and conventional approaches to information (signal) processing is that while conventional signal processing requires *a priori* knowledge of 'basis' functions, the given signal space

are essentially formulated as cascades of linear combinations of neurons and differ from isolated neurons, the basic processing units, in being able to represent a larger class of processors as compared to isolated neurons (see § 2.3).

Preservance of discrete spaces and the associated issue of function representation in neurons, as initiated in the previous chapter, leads, naturally, to an enquiry into the possibility of getting an insight into the aspect of function representation in layered neural signal processors. In this chapter, beginning with a study of representation in single layer neural signal processors – a network structure subjected to extensive investigations in the literature – I establish, assuming identical weights in all the processing nodes, that the number of distinct processing nodes needed to represent a function on a discrete space is bounded above, weakly, by the cardinality of the discrete space. The assumption of identicality in the weight vectors of distinct nodes is not unrealistic in view of the results established in Chapter 3.

I also establish that the issues of learning weight and threshold values and that of generalization in single layered neural signal processors

being identified as a subset of the linear span of the 'basis' functions, the neural signal processing approach seeks to synthesize the relevant 'basis' functions by identifying a suitable architecture and opting to choose the weight and threshold values of the various participating nodes in the architecture. The given signal space, as in conventional signal processing, is still sought to be identified as a subset of the linear span of the synthesized 'basis' functions; this particular interpretation, while analytically convenient, is not mandatory. 'Basis' function synthesis is a crucial component of the notion of representation in this thesis. The manner in which the 'basis' functions are synthesized in neural networks will be taken up in Chapter 6 relate to the corresponding issues in isolated neurons Representation of functions in single layered neural signal processors with the processing nodes having weights that form preservance weights for the (discrete) input space of the neural signal processor will be shown to reduce to a problem of function representation in single layered neural signal processors with identical weights in the processing nodes.

The reduction is facilitated by the symmetries and permutation between preservance weights discussed in the preceding chapter. Despite the fact that single layered neural signal processors have been extensively investigated, the notion of minimal architecture is conspicuously absent in the presentations of neural network based information processing. I propose a notion of minimal neural signal processing architectures, the criterion of minimality being related to that of admissibility of (preservance) weights introduced in § 3.3.

A study of the architecture of single layer neural signal processors shows that while these processors are able to represent all discrete functions on discrete spaces with hard-limiting activation function in the single layer of processing nodes and the space of functions represented is a dense subset of the space of continuous functions when sigmoidal activation function is employed in the processing nodes, the number of distinct processing nodes demanded to achieve the representation is unmanageably large. More precisely, the representational complexity is of exponential order.

The possibility of representing functions in multi-layered neural signal processors with a complexity smaller than that achievable in single layer has been studied; the study is, admittedly, of a preliminary nature in view of the analytical intractability introduced by the non-linear nature of the activation functions. In this study too, the notions of preservance of (discrete) input spaces and preservance weights are maintained to conform to the theme initiated earlier.

A preservation of the uniqueness and relative order between the input space points in the discriminants of the neurons by the weights associated with the discriminant (*ie* innerproduct) function is a characterization of the nature of 'internal' representations effected in neural networks. The issues of learning the weights of the first layer in a neural network (assumed to operate on a preservance input space) is essentially one of identifying a preservance input space to the collection of inputs described in the training set. I have suggested an approach that would aid an identification of a preservance input space given a training set.

Realization of discrete-valued processors on discrete spaces coupled with the interpretation ascribed to preservance input spaces encourages a study of the possibility of realizing mappings between symbolic spaces Algebraic properties being the only available characterization of symbol spaces, I establish an algebraic equivalent of the notion of linear separability: *a dichotomy over a symbol space, itself embedded* *in a semi-lattice, is linearly separable if each member of the partition induced by the dichotomy on the (input) symbol space is a semi-lattice.*

Neural signal processors with a single layer of decision making are studied in § 4.1: this study is a continuation of function representation initiated in the previous chapter. The representational issues in multilayered neural networks, in particular, the possibility of reducing the representational complexity through layering in feed-forward networks is taken up in § 4.2 (*p*. 201). § 4.3 (*p*. 208) focuses on identifying preservance input spaces appropriate to the collection of inputs in the training set. Symbolic computation in neural networks is discussed in § 4.4 (*p*. 218) in preparation for a study, in Chapter 5, of the abstract nature of the representational paradigm in neural networks.

4.1 Representation in Single Layer Neural Signal Processors

Neural signal processors with a single layer of decision making, as described in Chapter 2 (see § 2.3), are described, with a minor revision of notation, as

$$\mathfrak{y}(\underline{x}) \triangleq \sum_{i=1}^{m} v_i y_i(\underline{x}) - \theta = \sum_{i=1}^{m} v_i \sigma(\underline{w}_i \cdot \underline{x} - \theta_i) - \theta \tag{4.1}$$

In the above equation, $\underline{x}$ denotes the input patterns, $\underline{x} \in \Re^n$, m is the number of processing nodes in the (single) layer, $m = 1, 2, \ldots$; $\underline{w}_i$ are the weights associated with the processing nodes, $\underline{w}_i \in \Re^n$, $i = 1, 2, \ldots m$;

θ_i are the thresholds associated with the processing nodes, $\theta_i \in \Re$, $i = 1, 2, .. m$; σ denotes the activation function, v_i are the coefficients of the linear combination of neural decisions, $v_i \in \Re$, $i = 1, 2, .. m$, θ is the bias, $\theta \in \Re$, and $\mathfrak{y}$ is the neural signal processor response, $\mathfrak{y} \in \Re$.[2]

Given the relation between weights, in isolated neurons, and certain discrete spaces (of $\Re^n$) that are preserved entirely in a one-dimensional space (described by the chosen weight) as discussed in Chapter 3, it is of interest to investigate the representational potential of single layered neural signal processors under such preservance weights. As the discrete subset of $\Re^n$ preserved and the one-dimensional sub-space $\mathcal{L}_{\underline{w}_i}$ which accommodates the preservation points are decided in each neuron by the corresponding weights, $\underline{w}_i$, i being the index of the neuron, it is not feasible to carry out a general analytical discussion for all the situations of weight realization.

I will begin by considering the specific case wherein the weights of all nodes are identical, differences in the decision mechanism at the various nodes being provided by threshold values. Without any loss

[2] This approach, of using linear combinations of neural responses as the response of a processor is in vogue in the literature (see *eg*, **Rosenblatt**, 1958; **Albus**, 1975; **Hecht-Nielsen**, 1987b, 1987c; **Caudill & Butler**, 1990) and is also evident in the notion of instar–outstar neurons (**Grossberg**, 1982). It is important to note that though all the processing nodes employ the same activation function, the decisions at these nodes, based on the specific choice of weights and threshold, need not all be identical. Processing structures of the kind described in Equation 4.1 have been extensively investigated in the literature and, as stated earlier, have been shown (*cf*, **Cybenko**, 1990, **Hornik, Stinchcombe & White**, 1989) to represent the space of continuous functions with any desired accuracy.

of generality, this situation is considered as a discussion of the representation, by single layer neural signal processors, of functions defined over the discrete subset ${}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta}) \equiv {}^{w}_{2}\mathcal{P}^n_r(\zeta,\underline{\vartheta}) \subset \Re^n$ with appropriate admissible values for r, ζ and $\underline{\vartheta}$. (Note that the radix of numbering does not influence preservance except for accommodating a larger number of discrete points of $\Re^n$ as the radix increases.)

𝔓𝔯𝔬𝔭𝔬𝔰𝔦𝔱𝔦𝔬𝔫 **4.1.1** *If the weights of all processing nodes are identical in a single layer neural signal processor,* ie, $\underline{w}_\imath = \underline{w} \ \forall \imath, \ i = 1, 2, \quad m,$ *the component functions* $y_\imath$, $\imath = 1, 2, \ . \ m$, *are all defined on a common discrete subset* $\mathfrak{L}_{\underline{w}}(\|\underline{w}\|, {}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta}))$, *for all* $\underline{w} \in \Re^n$.

𝔓𝔯𝔬𝔭𝔬𝔰𝔦𝔱𝔦𝔬𝔫 **4.1.2** *Functions realized as* $\mathfrak{y}(\underline{x})$ *in a single layer neural signal processor wherein weights in all processing nodes are identical are a linear combination of the univariate functions, over* $\mathfrak{L}_{\underline{w}}$, *corresponding to the functions of the individual processing nodes.*

𝔓𝔯𝔬𝔭𝔬𝔰𝔦𝔱𝔦𝔬𝔫 **4.1.3** *Function realization in single layer neural signal processors wherein the weights of all processing nodes are identical is influenced only by the location of thresholds* θ_i, *coefficients of linear combination* $v_\imath$, $i = 1, 2, .. \ m$, *and bias* θ.

These statements follow immediately from Proposition 3.2.1 (*p*. 138) and the processing scheme introduced in Equation 4.1 (*p*. 188). Seeking the structure of functions realized over $\mathcal{L}_{\underline{w}}(\|\underline{w}\|, {}^{\underline{w}}\mathcal{P}_r^n(\zeta, \underline{\vartheta}))^3$ given a weight $\underline{w} \in \Re^n$, $\underline{w} \neq \underline{0}$, common to all processing nodes of a single layer neural signal processor, the following are easy to establish.

Proposition **4.1.4** *Sign transitions of each component function $y_\imath$, $\imath = 1, 2, \;.\; m$, under superposition induce level transitions in $\mathfrak{y}$.*

This statement is a direct consequence of $\mathfrak{y}(\underline{x})$ being realized as a linear combination of the component functions $y_\imath(\underline{x})$ for all $\underline{x} \in \Re^n$ and the addition of a bias θ.

Proposition **4.1.5** *The location of every level transition in $\mathfrak{y}(\underline{x})$ is inherited from a sign transition of $y_\imath(\underline{x})$ for some i, $\imath = 1, 2, \;.\; m$.*

Proposition **4.1.6** *Each sign transition of $y_\imath$, for all $\imath$, $\imath = 1, 2, ..\; m$, contributes to a single level transition of $\mathfrak{y}(\underline{x})$.*

Proposition 4.1.5 is a simple consequence of superposition and Proposition 4.1 6 follows from the notion of a function.

Theorem **4.1.1** *The number of level transitions in the discrete sequences over $\mathcal{L}_{\underline{w}}(\|\underline{w}\|, {}^{\underline{w}}\mathcal{P}_r^n(\zeta, \underline{\vartheta}))$, for any $\underline{w} \in \Re^n$ and admissible values*

[3] Note that the discrete space ${}^{\underline{w}}\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ consists of unions of binary collections, each of which is preserved by the (non-null) weight $\underline{w} \in \Re^n$ in $\mathcal{L}_{\underline{w}}$. As the preservance weight $\underline{w}$ is specific to this preservance input space, not only in direction, but also in magnitude, the collection of projection points, $\mathcal{L}_{\underline{w}}$, are indicated to be influenced by the norm $\|\underline{w}\|$ of the preservance weight $\underline{w}$.

of n, r, ζ and $\underline{\vartheta}$, resulting from a realization of single layer neural signal processors as a linear combination of m, $m = 1, 2,$, decision elements based on sigmoidal (including hard-limiter) activation functions and identical weights in all processing nodes does not exceed the number of component decision functions, ie, *m.*

This statement follows from Proposition 4.1.5 and Proposition 4.1.6. Note that the number of level transitions is independent of the bias θ and $\|\underline{w}\|$. Immediate implications of this theorem follow.

THEOREM **4.1.2** *A two layered network of neurons constructed as*

$$y^{(2)}(\underline{x}) = \sigma(\mathfrak{y}(\underline{x}))\,,\ \textit{for all}\ \underline{x} \in {}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta}) \subset \Re^n,\ \textit{for any}\ \underline{w} \in \Re^n$$

*and all admissible values of r, ζ and $\underline{\vartheta}$, exhibits no more than m sign transitions in the discrete sequences over $\mathfrak{L}_{\underline{w}}(\|\underline{w}\|\,, {}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta}))$ representing bipolar bivalent functions (*ie, *bipolar dichotomies) over ${}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ when the activation function σ is sigmoidal.*

THEOREM **4.1.3** *No more than $|{}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})| \equiv |\mathcal{P}^n_r(\zeta,\underline{\vartheta})|$ linearly separable nodes are necessary in the two layered neural network of* Theorem 4.1.2 *to realize all bipolar bivalent functions over ${}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$.*

This statement, however, is not unknown in the specific context wherein $r = 1$, $\zeta = 1$ and $\underline{\vartheta} = \underline{0}$, *ie*, $\mathcal{P}^n_1(1,\underline{0}) \triangleq \mathcal{B}^n$: bipolar bivalent functions (over $\mathcal{P}^n_1(1,\underline{0})$) are termed Boolean functions of n binary variables. For example, **Lippmann** (1987) and **Hertz, Krogh & Palmer** (1991) have given different geometric arguments in support of the claim that a linear combination of 2^n linear separable nodes followed by a hard-limiting comparator is adequate to realize all Boolean functions of n binary variables.

THEOREM **4.1.4** *Neural signal processors described by the functional form in* Equation 4.1 *realize all complete dichotomies on* ${}^{\underline{w}}\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ *provided sufficiently many processing nodes are available.*

PROOF: Consider a single layer neural signal processor given by:

$$\begin{aligned}\mathfrak{y}(\underline{x}) &= \sum_{i=1}^{|{}^{\underline{w}}\mathcal{P}_r^n(\zeta,\underline{\vartheta})|-1} v_i y_i(\underline{x}) - \theta,\\ y_i(\underline{x}) &= \sigma(\underline{w}\;\underline{x} - \theta_i),\end{aligned}$$

where, $\theta_i \in {}^{\underline{w}}\Theta_r^n(||\underline{w}||, \zeta, \underline{\vartheta})(i)$, $i = 1, 2, \ldots |{}^{\underline{w}}\mathcal{P}_r^n(\zeta,\underline{\vartheta})| - 1$. In this structure, which assumes the maximum number of processing nodes, the effective contribution of processing nodes representing trivial functions, *ie*, functions that are constant for all input values, is assumed to be specified by the bias term θ.

Noting that for each i, y_i represents a linear separable dichotomy on ${}^{\underline{w}}\mathcal{P}_r^n(\zeta,\underline{\vartheta})$ and the ordering imposed in the assignment to θ_i, in the sense that $\theta_1 < \theta_2 < \;.\;.\theta_{|{}^{\underline{w}}\mathcal{P}_r^n(\zeta,\underline{\vartheta})|-1}$, ensures that $\mathfrak{y}$ is expressed as a linear combination of all distinct (except for complements) linearly separable dichotomies on ${}^{\underline{w}}\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, given any dichotomy, expressed for notational convenience as the array $\mathfrak{y}$,[4]

$$\mathfrak{y} = \left[\mathfrak{y}(\underline{x}_\imath)\right]_{\imath=1}^{|{}^{\underline{w}}\mathcal{P}_r^n(\zeta,\underline{\vartheta})|}$$

such that

$$\underline{w}\,\underline{x}_\imath < \underline{w}\,\underline{x}_\jmath \text{ for all } \imath < \jmath,\ i, \jmath = 1, 2, \quad |{}^{\underline{w}}\mathcal{P}_r^n(\zeta, \underline{\vartheta})| - 1,$$

the functionality of the single layer neural signal processor can equivalently be expressed as:

$$\mathfrak{y} = \boldsymbol{y}\underline{v}, \tag{4.2}$$

where, $\underline{v} = \left[v_1, v_2, \ldots v_{|{}^{\underline{w}}\mathcal{P}_r^n(\zeta,\underline{\vartheta})|-1}\right]^{\mathsf{T}}$ and $\boldsymbol{y}$ is the matrix

$$\boldsymbol{y} = \left[y_\jmath\left(\underline{x}_\imath\right)\right]_{\imath=1}^{|{}^{\underline{w}}\mathcal{P}_r^n(\zeta,\underline{\vartheta})|}{}_{\jmath=1}^{|{}^{\underline{w}}\mathcal{P}_r^n(\zeta,\underline{\vartheta})|-1}.$$

The matrix $\boldsymbol{y}$ is independent of the given dichotomy and, hence, a solution to the linear system indicated in Equation 4.2, in an attempt to find the minimum norm solution for $\underline{v}$, would be

$$\underline{v} = \left(\boldsymbol{y}^{\mathsf{T}}\boldsymbol{y}\right)^{\star}\boldsymbol{y}^{\mathsf{T}}\mathfrak{y},$$

where, $\boldsymbol{A}^\star$ is the *Moore-Penrose* pseudo-inverse (*cf*, **Penrose**, 1955; **Ben-Israel & Greville**, 1974) of a matrix $\boldsymbol{A}$.

From the structure of $\boldsymbol{y}$ it is immediately evident that $\boldsymbol{y}^{\mathsf{T}}\boldsymbol{y}$ is non-singular and, hence, its Moore-Penrose pseudo-inverse is assured.

□

[4] $\mathfrak{y}$ is the equivalent sequence of the given dichotomy on $\mathcal{L}_{\underline{w}}$ for a non-null weight $\underline{w} \in \Re^n$.

Comparing the representation provided by network structures described by Equation 4.1 (*p*. 188) with that provided by a two layer neural network suggested in Theorem 4.1.2 (*p*. 192), it is immediately apparent that the activation function, operating on the response $\mathfrak{y}$, serves to render the bias term θ of neural signal processors redundant.

Though the above statement has been established with the notion of sufficiently many processing nodes interpreted in the sense of a finite upper bound on the number of processing nodes, it is never the case that all processing nodes, each representing distinct linearly separable bipolar bivalent functions over ${}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, will participate in the synthesis of every dichotomy on ${}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$. Noting from Proposition 4.1.5 (*p*. 191) and Proposition 4.1.6 (*p*. 191) that the number of decision elements in a single layer neural signal processor is identical to the number of sign-transitions in $\mathfrak{y}$, the number of processing nodes required to represent a given dichotomy can be made smaller than $|{}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})| - 1$ by an appropriate choice of common preservance weight $\underline{w}$. This consideration, being similar to the notion of admissibility of weights to a given dichotomy, together with the notion of architecture discussed in Chapter 2 (see § 2 2) prompts the following.

DEFINITION 4.1.1 *Given a function f: ${}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta}) \rightarrow [\zeta_-,\zeta_+]$, or dichotomy $\mathfrak{D}$· ${}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta}) \rightarrow \{\zeta_-,\zeta_+\}$, for any non-null weight $\underline{w} \in \Re^n$, the architecture of a single layer neural signal processor, expressed as the combined specification of the number of nodes m in the single layer* *of processing and weights and thresholds in the individual processing nodes, $\underline{w}_i$, θ_i, $i = 1, 2, \ldots m$, bias θ and co-efficients of linear combination $\underline{v}$ that realize $f\ (\mathfrak{D})$, is termed* minimal *for $f\ (\mathfrak{D})$ if a realization of $f\ (\mathfrak{D})$ in that architecture cannot be achieved with fewer than m processing nodes.*

Having established the nature of relationship between the function $\mathfrak{y}$ realized by a single layer neural signal processor and the component (decision functions) y_i, $i = 1, 2, \ldots m$, in the restricted context of all processing nodes having identical weights, it is imperative that attention be given to the crucial problem of learning, *ie*, an automated specification of values for the common weights $\underline{w}$, threshold θ_i, superposition co-efficients v_i, $i = 1, 2, \ldots m$, and bias θ In view of the fact that $\mathfrak{y}$ is a point-wise addition of component functions y_i, $i = 1, 2, \ldots m$, over the support $\mathfrak{L}_{\underline{w}}(\|\underline{w}\|, {}^{\underline{w}}\mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ as suggested in Proposition 4 1.1 (*p*. 190) and Proposition 4.1.2 (*p*. 190), specification of the weight $\underline{w}$ will involve the admissibility criterion discussed in § 3.3 (see Theorem 3.3.2 (*p*. 161)). However, this criterion will be applicable only over the component processing nodes and not on the overall function synthesized by the neural signal processor as a consequence of Theorem 4.1.4 (*p*. 193).

Learning (with generalization), of minimal architectures, in view of the preceding definition is formulated as the following search problem:

$$\min_{\{\underline{w}_i\}_i, \{\underline{\theta}_i\}_i, \theta, \underline{v}} \sum_{i=1}^{|{}^{\underline{w}}\mathcal{P}_r^n(\zeta, \underline{\vartheta})| - 1} |f(\underline{x}_{i+1}) - f(\underline{x}_i)| \,. \tag{4.3}$$

In the above expression, ordering in the points of ${}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ is in the same sense as discussed in the proof of Theorem 4.1.4 (p. 193), f is the desired function on ${}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, available through examples, that is to be represented in the single layer neural signal processor and the collections $\{\underline{w}_j\}$ and $\{\underline{\theta}_j\}$ are with respect to the m processing nodes in the ensemble.

In addition to the above criterion, the search is based on a criterion that evaluates the *goodness* of approximation in the sense of an appropriately designed norm on the space of training samples. This term, however, has not been indicated in Equation 4.3 as Theorem 4.1.4 (p. 193) assures exactness of realization particularly when hard-limiting activation functions are used. In light of the discussion, in Chapter 3, the procedure for search of function representation with preservance weights and on preservance input spaces is of the following nature.

THEOREM **4.1.5** *Learning of a given non-trivial bipolar dichotomy* $\mathfrak{D}.\ \mathcal{T}_\iota \to \{\zeta_-,\zeta_+\}$ *in a single layer neural signal processor with* m *processing nodes and hard-limiting activation function involves the sequence of three distinct steps:*

1. *Identify, for a given* $\epsilon > 0$, *a suitable discrete subset* ${}^{\underline{w}_0}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, $\underline{w}_0 \in \Re^n$, $\|\underline{w}_0\| \neq 0$, $r = 1, 2, \ldots$, $\zeta \in \Re_+$ *and* $\underline{\vartheta} \in \Re^n$ *corresponding to* $\mathcal{T}_\iota \subset \Re^n$, $|\mathcal{T}_\iota| < \infty$, *such that*

$$\max_{x_\iota \in \mathcal{T}_\iota} \quad \min_{x \in {}^{\underline{w}_0}\mathcal{P}^n_r(\zeta,\underline{\vartheta})} \quad |\underline{x}_\iota - \underline{x}| \leq \epsilon$$

Assign to the weight $\underline{w}$ any element in ${}^{\underline{w}_0}\wp_n$, and correspondingly assign to m, the number of processing nodes in the single layer neural signal processor, the number of level transitions in the (equivalent) discrete sequence over $\mathfrak{L}_{\underline{w}_0}(\|\underline{w}_0\|, {}^{\underline{w}_0}\mathcal{P}^n_r(\zeta, \underline{\vartheta}))$ representing the given dichotomy $\mathfrak{D}$.

2. *Order the locations of the level transitions, and for $i = 1, 2, .. m$ assign the threshold θ_i any element of ${}^{\underline{w}_0}\Theta(\|\underline{w}_0\|, \zeta, \underline{\vartheta})(j)$, for an appropriate value of j, $j = 1, 2, .\ \ 2(|{}^{\underline{w}_0}\mathcal{P}^n_r(\zeta, \underline{\vartheta})| - |{}^{\underline{w}_0}\mathcal{P}^n_{r-1}(\zeta, \underline{\vartheta})| - 1)$, where ${}^{\underline{w}_0}\Theta(\|\underline{w}_0\|, \zeta, \underline{\vartheta})(j)$ contains the location of the ith level transition.*

3. *Assignments to the superposition coefficients v_i, $i = 1, 2, .\ .m$, and bias θ follow the minimum norm solution to a system of linear equations:*

$$\begin{bmatrix} \underline{v} \\ \theta \end{bmatrix} = \left(\tilde{\boldsymbol{y}}^{\mathsf{T}} \tilde{\boldsymbol{y}}\right)^{\star} \tilde{\boldsymbol{y}}^{\mathsf{T}} \underline{\mathfrak{y}},$$

where,

$$\tilde{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y} & \underline{1} \\ \underline{1}^{\mathsf{T}} & 1 \end{bmatrix}, \boldsymbol{y} = [y_i(\underline{x}_j)]_{i,j}, \ \underline{\mathfrak{y}} = [\mathfrak{y}(\underline{x}_j)]_j,$$

$i = 1, 2, \ldots m$, $j = 1, 2, \ldots |{}^{\underline{w}_0}\mathcal{P}^n_r(\zeta, \underline{\vartheta})|$,

and $\boldsymbol{A}^{\star}$ is the Moore-Penrose pseudo inverse of the matrix $\boldsymbol{A} \in \Re^{m \times n}$.

The approximation in step 1 of the above theorem can always be assured in view of the denseness of the discrete space ${}^{\underline{w}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ in the interval $\zeta\left[-\alpha\mathfrak{r}^{n-1}-\frac{1}{(\mathfrak{r}^n-1)},\alpha\mathfrak{r}^{n-1}+\frac{1}{(\mathfrak{r}^n-1)}\right]+\underline{w}\cdot\underline{\vartheta}$, where the scale factor $\alpha=\|\underline{w}\|\,(\sum_{\imath=1}^{n}\mathfrak{r}^{2(\imath-1)})^{-1}$, as r tends to $+\infty$. (Note that α is the scale factor of the (non-null) weight $\underline{w}$ in relation to a preservance weight of the discrete space ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$.) The steps involved in the identification of the preservance input space ${}^{\underline{w}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ will be discussed in § 4.3 (p. 208). The above theorem is more in the nature of a statement assuring the existence of a representation.

As seen in the case of admissibility of preservance weights to a given dichotomy, minimal architectures for a given dichotomy are not unique. The chief reason for such a non-unique representation in neural signal processors can easily be traced to Proposition 3.2.13 (p. 152). (Note that non-uniqueness in the sense of permutations of the co-efficients $\underline{w}$ with a corresponding permutation of weights and thresholds of the processing nodes are not being considered.) It is of interest to note that non-uniqueness in the minimal single layer neural signal processing architecture, due to the multiplicity of preservance weights representing a given dichotomy even at the level of the equivalent sequences over $\mathcal{L}_{\underline{w}}$, $\underline{w}$ being a non-null weight in $\Re^n$, cannot be resolved by the criterion of generalization.

Representation of dichotomies in single layer neural signal processors, discussed till now in the restricted case wherein weights of all processing nodes are identical, when extended to a situation permitting processing node weights to be members of ${}^{\underline{w}}\wp_n$ subject to the restriction that all processing nodes employ weights of identical norm implies the following.

$\mathcal{THEOREM}$ **4.1.6** *A problem of representing a given bipolar bivalent function* $\mathfrak{D}\cdot\ {}^{\underline{w}}\mathcal{P}^{n}_{r}(\zeta,\underline{\vartheta}) \rightarrow [\zeta_-,\zeta_+]$, *for some non-null* $\underline{w} \in \Re^n$, *in a single layer neural signal processor with* m *processing nodes,* $m < n!2^n$, *and weights* $\underline{w}_\imath \in {}^{\underline{w}}\wp_n$, $\imath = 1,2,\ldots m$, *such that* $\|\underline{w}_\imath\| = \|\underline{w}_m\|$, $i = 1,2, \quad ,m-1$, *is equivalent to the problem of representing* $\mathfrak{D}$ *in a single layer neural signal processor of* m *nodes wherein the weights of all processing nodes are identical, the common weight being any of* $\underline{w}_\imath$, $\imath = 1,2, \ . \ ,m$.

$\mathcal{PROOF}$: Noting that

a) the collection of weights described by ${}^{\underline{w}}\wp_n$, $\|\underline{w}_\imath\| = \|\underline{w}_1\|$, $i = 2,3,\ldots|{}^{\underline{w}}\wp_n|$, a space isomorphic to $\wp_n(\alpha)$, for any $\alpha \in \Re_+$, as indicated in § 3.4, are described in terms of any specific weight, in that space, through permutations similar to those indicated in Theorem 3.1.4 (*p*. 124), and

b) the procedure for determining $\underline{v}$, the co-efficients of linear combination of the responses of the m distinct neurons is not dependent on $y_\imath$, $i = 1,2,\ldots m$, being linearly separable,

the statement is immediately apparent.

□

4.2 Representation in Multi Layer Neural Signal Processors

Single layered neural signal processors, shown to be adequate in representing multivariate functions by **Hornik, Stinchcombe & White** (1989), **Cybenko** (1990) and in the preceding discussion (the discussion was restricted to the realization of bipolar dichotomies on preservance input spaces), suffer from the requirement that the number of processing nodes be exponentially dependent on the dimensionality of the input space as well as the number of disconnected components in the inverse images of the dichotomy under consideration. This exponential dependence is directly related to the number of allowed sign transitions in the sequences over $\mathfrak{L}_{\underline{w}}(\|\underline{w}\|, {}^{\underline{w}}\mathcal{P}_r^n(\zeta, \underline{\vartheta}))$, for any $\underline{w} \in {}^{\underline{w}_0}\wp_n$, $\|\underline{w}\| = \|\underline{w}_0\|$, with $\underline{w}_0 \in \Re^n$, $\|\underline{w}_0\| \neq 0$, and inputs restricted to the preservance input space ${}^{\underline{w}}\mathcal{P}_r^n(\zeta, \underline{\vartheta}) \subset \Re^n$ for admissible values of r, ζ and $\underline{\vartheta}$.

Representation of functions in single layer neural signal processors will exhibit such an exponential dependence regardless of the nature of activation functions which accommodate fixed finite number of sign transitions. It is thereby imperative to investigate approaches that would aid a reduction in the representational complexity of dichotomies. The representational complexity is to be interpreted as the number of distinct basic processing nodes that need to committed to achieve the desired representation relative to the number of similar (identical) nodes required in a minimal single layer neural signal processor.

Multi layered neural signal processors of the strictly feed-forward variety, in the light of Equation 4.1 (*p.* 188), are defined by the following operational schema (as distinct from the definition indicated in § 2.2):

$$
\begin{aligned}
y^{(1)}_{j^{(1)}}(\underline{x}) &= \sigma(\underline{w}^{(1)}_{j^{(1)}}\, \underline{x} - \theta^{(1)}_{j^{(1)}})\,, j^{(1)} = 1,2,\ldots m_1, && (4.4a)\\
y^{(\ell)}_{j^{(\ell)}}(\underline{x}) &= \sigma(\underline{w}^{(\ell)}_{j^{(\ell)}}\, \underline{y}^{(\ell-1)}(\underline{x}) - \theta^{(\ell)}_{j^{(\ell)}})\,, \\
& \qquad\qquad j^{(\ell)} = 1,2,.\ .m_\ell,\ \ell = 1,2,.\ \ k, && (4.4b)\\
\mathfrak{y}^{(k)}(x) &= \sum_{j=1}^{m_k} v_j y_j^{(k)}(\underline{x}) - \theta, && (4.4c)
\end{aligned}
$$

for some *a priori* specified values of k and m_ℓ, $k, m_\ell = 1, 2, \quad ; \ell = 1, 2, \ldots k$. In this equation, as already indicated in Chapter 2, $\underline{w}^{(\ell)}_{j^{(\ell)}}$ is the weight, $\theta^{(\ell)}_{j^{(\ell)}}$ is the threshold, and $y^{(\ell)}_{j^{(\ell)}}$ indicates the response of a processing node (*ie*, neuron) in layer ℓ, the node index being $j^{(\ell)}$, and $\mathfrak{y}^{(\ell)}$ is the response of a neural signal processor formed by linearly combining the outputs of a neural network of ℓ layers.

In view of Theorem 4.1.6 (*p.* 200) it is not difficult to visualize that the response of a two layer neural signal processor wherein the weights in the processing nodes of the first layer belong to the preservance weight space ${}^{\underline{w}_0}\wp_n$, subject to the restriction that $||\underline{w}^{(1)}_{j^{(\ell)}}|| = ||\underline{w}_0||$, $j^{(\ell)} = 1, 2, \ldots m_1$, $\underline{w}_0 \in \Re^n$, $||\underline{w}_0|| \neq 0$, is indeed equivalent to a superposition of m_2 functions, each of which is a bipolar bivalent sequence on the discrete space $\mathfrak{L}_{\underline{w}_0}(||\underline{w}_0||\,, {}^{\underline{w}_0}\mathcal{P}^n_r(\zeta, \underline{\vartheta}))$. (Note that $\mathfrak{L}_{\underline{w}_0}(||\underline{w}_0||\,, {}^{\underline{w}_0}\mathcal{P}^n_r(\zeta, \underline{\vartheta})) \equiv \mathfrak{L}_{\underline{w}}(||\underline{w}||\,, {}^{\underline{w}}\mathcal{P}^n_r(\zeta, \underline{\vartheta}))$ as ${}^{\underline{w}_0}\mathcal{P}^n_r(\zeta, \underline{\vartheta}) \equiv {}^{\underline{w}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ for all $\underline{w} \in {}^{\underline{w}_0}\mathcal{P}_n$ such that $||\underline{w}|| = ||\underline{w}_0||$.) The bipolar bivalent sequences (on $\mathfrak{L}_{\underline{w}_0}(||\underline{w}_0||\,, {}^{\underline{w}_0}\mathcal{P}^n_r(\zeta, \underline{\vartheta}))$) are responses of appropriate single layer neural signal processors and the number of distinct sign transitions in each of these sequences is no more than m_1, the number of processing nodes in the first layer; this observation results as a simple consequence of Theorem 4 1.1 (*p*. 191).

The following characterization, in view of the above mentioned feature, about the equivalent sequences representing functions realized by two layer neural signal processors is worth noting.

THEOREM **4.2.1** *The number of distinct level transitions in the discrete sequences over* $\mathfrak{L}_{\underline{w}_0}(\|\underline{w}_0\|, {}^{\underline{w}_0}\mathcal{P}^n_r(\zeta,\underline{\vartheta}))$ *for any* $\underline{w}_0 \in \Re^n$, $\|\underline{w}_0\| \neq 0$, $n = 1, 2, \ldots$; $r = 1, 2, \ldots$; $\zeta \in \Re_+$ *and* $\underline{\vartheta} \in \Re^n$, *resulting from a two layer neural signal processor wherein the weights corresponding to processing nodes in the first and second layers are given by* $\underline{w}^{(1)}_{j^{(1)}} \in {}^{\underline{w}_0}\wp_n$, $\|\underline{w}^{(1)}_{j^{(1)}}\| = \|\underline{w}_0\|$, $j^{(1)} = 1, 2, \ldots m_1$, *and* $\underline{w}^{(2)}_{j^{(2)}} \in \Re^n$, $j^{(2)} = 1, 2, \ldots m_2$, *is bounded above by*

$$\min(m_2 m_1, |{}^{\underline{w}_0}\mathcal{P}^n_r(\zeta,\underline{\vartheta})| - 1)$$

Since this statement is on the same lines as Theorem 4.1.1 (*p*. 191), a proof is not required. A generalization of the above theorem to the case of multi-layered neural networks follows.

THEOREM **4.2.2** *The number of distinct level transitions in the discrete sequences over* $\mathfrak{L}_{\underline{w}_0}(\|\underline{w}_0\|, {}^{\underline{w}_0}\mathcal{P}^n_r(\zeta,\underline{\vartheta}))$ *for any* $\underline{w}_0 \in \Re^n$, $\|\underline{w}_0\| \neq 0$, $n = 1, 2, \ldots$; $r = 1, 2, \ldots$; $\zeta \in \Re_+$ *and* $\underline{\vartheta} \in \Re^n$, *resulting from a* k *layer neural signal processor,* $k = 1, 2, \ldots$, *wherein the weights corresponding to processing nodes in the various layers are given by* $\underline{w}^{(1)}_{j^{(1)}} \in {}^{\underline{w}_0}\wp_n$,

$\|\underline{w}_{j^{(1)}}^{(1)}\| = \|\underline{w}_0\|$, $j^{(1)} = 1, 2, \ldots m_1$, *and* $\underline{w}_{j^{(\ell)}}^{(\ell)} \in \Re^n$, $j^{(\ell)} = 1, 2, \ldots m_\ell$, $\ell = 2, 3, \;..k$, *is bounded above by*

$$min(\prod_{\ell=1}^{k} m_\ell, |^{\underline{w}_0}\mathcal{P}_r^n(\zeta, \underline{\vartheta})| - 1)$$

The above two theorems suggest that a study of representation in multi-layered neural signal processors is analogous to a corresponding study in single layer neural signal processors, though with processing nodes that incorporate activation functions that induce multiple level (sign) transitions. An immediate implication is that the approach to learning suggested by Theorem 4.1.5 (p. 197) and Theorem 4.1.6 (p. 200) is applicable, though with appropriate changes, to the case of multi-layered neural signal processors also.

Recall Equation 4.4c. While the procedure for determining the coefficients v_j, $j = 1, 2, .. \; m_k$, and bias θ in a k, $k = 1, 2, \; .$, layered neural signal processor is the same as that indicated in step 3 of Theorem 4.1.5, the matrix $\boldsymbol{y}$, in the case of multi-layered neural signal processors, consists of sequences over $\mathfrak{L}_{\underline{w}_0}(\|\underline{w}_0\|, {}^{\underline{w}_0}\mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ with multiple sign transitions. ($\underline{w}_0$ is assumed to be a preservance weight for the discrete preservance input space.) Note that these sequences correspond to neural responses of a (vector-valued) neural network of k layers (for a k layered neural signal processor). The sequences are representative of the bipolar bivalent (decision) functions over ${}^{\underline{w}_0}\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ that are linearly combined (with coefficients v_j, $j = 1, 2, \ldots m_k$) to realize (approximate) the desired function.

To facilitate a procedure for determining the values of the coefficients (v) of linear combination I will denote by ${}^{\mathfrak{z}_k}\boldsymbol{y}^{(k)}$ the collection of sequences (over $\mathfrak{L}_{\underline{w}_0}(\|\underline{w}_0\|, {}^{\underline{w}_0}\mathcal{P}^n_r(\zeta,\underline{\vartheta}))$) corresponding to the decisions of a k layered neural network with no more than $\mathfrak{z}_k$ sign transitions. Theorem 4.2.2 (p. 203) states an upper bound for the value of $\mathfrak{z}_k$. Symbolically,

$$\begin{aligned} {}^{\mathfrak{z}_k}\boldsymbol{y}^{(k)} &= \left[y^{(k)}_{\jmath^{(k)}}(\underline{x}_\imath)\right]_{\imath=1}^{|{}^{\underline{w}_0}\mathcal{P}^n_r(\zeta,\underline{\vartheta})|}{}_{\jmath^{(k)}=1}^{m_k} \\ &\qquad \text{such that } \forall j^{(k)} \sum_{\imath=1}^{|{}^{\underline{w}_0}\mathcal{P}^n_r(\zeta,\underline{\vartheta})|-1} \left|y^{(k)}_{\jmath^{(k)}}(\underline{x}_{\imath+1}) - y^{(k)}_{\jmath^{(k)}}(\underline{x}_\imath)\right| \le \mathfrak{z}_k \end{aligned}$$

Denote by the array $\mathfrak{y}^{(k)}$ the following:

$$\mathfrak{y}^{(k)} = \left[\mathfrak{y}^{(k)}(\underline{x}_\imath)\right]_{\imath=1}^{|{}^{\underline{w}_0}\mathcal{P}^n_r(\zeta,\underline{\vartheta})|}.$$

Then, in a manner analogous to step 3 in Theorem 4.1.5 (p. 197), a solution for the coefficients of linear combination (v) and the bias (θ) is given by

$$\begin{bmatrix} \underline{v} \\ \theta \end{bmatrix} = \left(\tilde{\boldsymbol{y}}^{\mathsf{T}}\tilde{\boldsymbol{y}}\right)^{*}\tilde{\boldsymbol{y}}^{\mathsf{T}}\underline{\mathfrak{y}}^{(k)}, \text{ where } \tilde{\boldsymbol{y}} = \begin{bmatrix} {}^{\mathfrak{z}_k}\boldsymbol{y}^{(k)} & \underline{1} \\ \underline{1}^{\mathsf{T}} & 1 \end{bmatrix}$$

Note that given a value of $\mathfrak{z}_k$, subject to the upper bound suggested by Theorem 4.2.2, the matrix ${}^{\mathfrak{z}_k}\boldsymbol{y}^{(k)}$ is completely specified for the (discrete) preservance input space ${}^{\underline{w}_0}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$. This matrix is not dependent on the specific values assigned to the coefficients of linear combination (v) and threshold (θ). An immediate implication is that each row of ${}^{\mathfrak{z}_k}\boldsymbol{y}^{(k)}$, a sequence over $\mathfrak{L}_{\underline{w}_0}(\|\underline{w}_0\|, {}^{\underline{w}_0}\mathcal{P}^n_r(\zeta,\underline{\vartheta}))$, states the collection of assignments, over ${}^{\underline{w}_0}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, for a distinct node in the k-th layer.

Recall Proposition 4.1.5 and Proposition 4.1.6. In the context of multi-layered neural signal processors, these propositions state the dependence of the number and location of level transitions in the response[5] $\mathfrak{y}^{(\ell)}$ on the (number and) location of sign transitions in the decisions $y^{(\ell)}$ for all values of ℓ, $\ell = 1, 2, \ldots k$. (Note that the response $\mathfrak{y}^{(\ell)}$, for $\ell \neq k$ is defined, analogous to Equation 4.4c, as a shifted linear combination of the decisions $y^{(\ell)}_{j^{(\ell)}}$, $j^{(\ell)} = 1, 2, \ .. m_\ell$.)

On reverting back to the situation wherein the matrix ${}^{\mathfrak{z}_k}y^{(k)}$ is completely known given $\mathfrak{z}_k$ and ${}^{\underline{w}_0}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$, it is of interest to know the converse of Proposition 4.1.5 and Proposition 4.1.6, *ie*, the dependence of the level transitions, in number and location, of the decisions $y^{(\ell+1)}$ on the (number) and location of sign-transitions in the responses $\mathfrak{y}^{(\ell)}$, $\ell = 1, 2, . \ . k-1$. The converses are similar in nature to Proposition 4.1.5 and Proposition 4.1.6, *ie*, the location of every sign transition in $y^{(\ell+1)}_{j^{(\ell+1)}}$, for all $j^{(\ell+1)} = 1, 2, \ \ . m_{\ell+1}$, is inherited from a level-transition of $\mathfrak{y}^{(\ell)}_{j^{(\ell)}}$, for some $j^{(\ell)} = 1, 2, \ .. m_\ell$, and every level transition in $\mathfrak{y}^{(\ell)}_{j^{(\ell)}}$, for all $j^{(\ell)} = 1, 2, .. \ m_\ell$, contributes to at most one sign-transition in $y^{(\ell+1)}_{j^{(\ell+1)}}$, for some $j^{(\ell+1)} = 1, 2, \ldots m_{\ell+1}$.

Thus, a knowledge of the matrix ${}^{\mathfrak{z}_k}y^{(k)}$ provides an insight into the locations of level crossings in the responses $\mathfrak{y}^{(k-1)}$ of the $k-1$-th layer. A repeated application of the above approach to elicit the locations of

[5] References, in this section, to $\mathfrak{y}$ and y, with appropriate layering and node indices, are to be interpreted as the equivalent sequences corresponding to these functions on the discrete space $\mathfrak{L}_{\underline{w}_0}(\|\underline{w}_0\|, {}^{\underline{w}_0}\mathcal{P}^n_r(\zeta, \underline{\vartheta}))$

level transitions in the responses of the $\ell - 1$-th layer with a knowledge of the locations of level transitions in the ℓ-th layer suggests a scheme which is analogous to the *back-propagation* procedure common in the literature. The specific details of the scheme of back-calculation of the locations of level transitions are not presented in this preliminary discussion. Each back-calculation step necessitates a foreknowledge (or a judicious selection) of the largest number of sign transitions that will be tolerated in the decisions of the corresponding layers.

From the preceding discussion it is amply clear that it is not impossible to realize, in a multi-layer neural signal processor, a given dichotomy on ${}^{\underline{w}_0}\mathcal{P}_r^n(\zeta,\underline{\vartheta})$, $\underline{w}_0 \in \Re^n$, $\|\underline{w}_0\| \neq 0$, and admissible values for n, r, ζ and $\underline{\vartheta}$, with fewer total processing nodes than necessary in a single layer neural signal processor as given any integer, say m, $m = 1, 2, \ldots$, it is not infeasible to choose k integers $m_1, m_2 \ldots m_k$, $m_\ell = 1, 2, \; .. m - 1$, $\ell = 1, 2, \ldots k$, $k = 2, 3, \ldots$, such that $\prod_{\ell=1}^{k} m_\ell = m$ whereas $\sum_{\ell=1}^{k} m_\ell < m$. The reduction in the number of processing nodes required in a multi-layered neural signal processor relative to the requirement in a minimal single layer neural signal processor can easily be traced to the multiplicity of the sign-transitions in the bipolar bivalent sequences (over $\mathfrak{L}_{\underline{w}_0}(\|\underline{w}_0\|, {}^{\underline{w}_0}\mathcal{P}_r^n(\zeta,\underline{\vartheta})))$ representing functions (dichotomies) over ${}^{\underline{w}_0}\mathcal{P}_r^n(\zeta,\underline{\vartheta})$: in the limited scope of this thesis, this aspect will not, however, be elaborated.

4.3 Learning of Weights: Identification of preservance input spaces

In Chapter 3, the representation of functions (on discrete spaces) in isolated neurons been shown to be influenced, in the sense of a simplification, by a choice of weights that are related to the (discrete) input space: the specific relation considered is that of preservance. Analogous to the representation of functions in isolated neurons, the representation of functions (processors) in multi-layered neural signal processors is influenced by a choice of preservance weights in the first layer.[6]

The role of preservance, as suggested by the definition, is to maintain uniqueness and orderings between the distinct points of the input space: such a uniqueness can, however, be established only when the input space is discrete. In multi-layered neural signal processors, a choice of preservance weights is tantamount to a representation of the input space. With this interpretation, it is not incorrect to suggest that the processing in isolated neurons, and their ensembles, involves an internal representation of the (relevant) input space and the processor realization is based on this internal representation.

In this section I will consider the issue of identifying a discrete space given $\mathcal{T}_i$, the inputs contained in a training set, such that the

[6] While it is not infeasible to discuss a situation wherein the weights of each layer, in a multi-layered neural signal processor, are preservance weights corresponding to the inputs space presented to that layer, such an investigation has not been attempted in this thesis.

discrete space is a preservance input space corresponding to some non-null weight vector in $\Re^n$. Equivalently, the problem is one of specifying the weight of an isolated neuron (or the common weight in the first layer of a multi-layered neural signal processor) such that a discrete input space which embeds an approximation (in the sense of the Euclidean norm) of $\mathcal{T}_i$ is preserved by the chosen weight.

Recall that preservance input spaces are denoted by $\frac{\underline{w}}{\mathfrak{r}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$. In this notation n is the dimensionality of the Euclidean space in which this discrete space is embedded. $\mathfrak{r}$ is the radix of numbering, *ie*, the number of distinct (discrete) values that are allowed in each element of the n-vectors (*ie*, n-tuples of numbers) in the input space. r refers to the degree of ranking. ζ is the common scale factor and $\underline{\vartheta}$ is the common translation that every point in the input space are subjected to· the scale and translation are measured in relation to the elements of $\mathcal{B}^n$ the collection of binary vectors in $\Re^n$. $\underline{w}$ indicates the 'direction' (*ie*, $\frac{\underline{w}}{\|\underline{w}\|}$) of a preservance weight[7] of the discrete collection of points in $\Re^n$. An identification of a preservance input space given $\mathcal{T}_i$ involves a specification of all the above components.

In the following discussion, I assume that the dimensionality n of the Euclidean space from which the training inputs are drawn is known *a priori*. A few comments are in order about the nature of the discrete set $\mathcal{T}_i$ in relation to the discrete space $\frac{\underline{w}}{\mathfrak{r}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$, for some appropriately

[7] Note that there are $n!\,2^n$ distinct preservance directions for every n dimensional (preservance) input space

identified values of $\underline{\vartheta}$, ζ, $\mathfrak{t}$, r and $\underline{w}$ (the dimensionality n is given). Recall the construction of the discrete space ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ in § 3.4 (Equation 3.11 (p. 179)). For any non-null $\underline{w} \in \Re^n$, ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ is equivalent to a 'rotation' (actually permutation) of ${}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{0})$ followed by a translation.

The operator permuting (rotating) ${}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{0})$ to ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{0})$ is governed by the linear transformation that maps to the weight $\underline{w}$ the preservance weight $\underline{w}_{<\mathfrak{e}>}$, for an appropriate value of $\mathfrak{e}$, $\mathfrak{e} = 0, 1, \ldots n!2^n - 1$, that is closest, in the sense of the Euclidean distance, to $\underline{w}$ in ${}_{\mathfrak{t}}\mathcal{S}_n$, the class of preservance weights of ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{0})$, subject to the restriction that $\|\underline{w}_{<\mathfrak{e}>}\| = \|\underline{w}\|$. A translation of all points in ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{0})$ by $\underline{\vartheta}$ results in the collection denoted by ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$.

Note that the discrete space ${}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{0})$ is obtained as a union of certain scaled and translated collections of binary vectors: the scale and translation factors operating on the component binary vector collections are described in Equation 3.7 (p. 172). Each collection of binary vectors, denoted by $\mathcal{B}^n(\zeta,\underline{\vartheta})$, for a scale factor $\zeta \in \Re_+$ and a translation $\underline{\vartheta} \in \Re^n$, has the algebraic properties of a lattice (Boolean algebra)[8] under a suitably identified pair of binary operations operating on the collection.[9]

Though not all unions of lattices (algebras) result in algebraic structures that are lattices (algebras), the specific scale factor and transla-

[8] For this reason the term Boolean space for the collection of binary vectors $\mathcal{B}^n$ and its generalization $\mathcal{B}^n(\zeta,\underline{\vartheta})$ is not inappropriate

[9] The most common place examples are the logical operations of conjunction and disjunction.

tions chosen in the construction of ${}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{0})$ ensure that this discrete space is also a lattice (algebra): this aspect will be considered, more formally, in the next section. Since ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ is obtained from ${}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{0})$ by a norm preserving permutation[10] followed by a translation, the algebraic characteristics of ${}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{0})$ are inherited by ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$. This implies that ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ has the algebraic structure of a lattice (algebra), however, it is not essential that binary operations on ${}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{0})$ be inherited identically by ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$.

A specification of a weight $\underline{w}$ in an isolated neuron such that $\underline{w}$ preserves $\mathcal{T}_{\mathfrak{t}}$ in $\mathcal{L}_{\underline{w}}$ implies that with appropriate values for $\mathfrak{t}$, r, ζ and $\underline{\vartheta}$, the preservance, by weight $\underline{w}$, is applicable not only to the points in the discrete set $\mathcal{T}_{\mathfrak{t}}$ but to the entirety of the discrete space ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$. Note that for the specification of the weight and other parameters to be meaningful, $\mathcal{T}_{\mathfrak{t}} \subseteq {}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$. As the discrete space ${}^{\underline{w}}_{\mathfrak{t}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, given $\underline{w}$, $\mathfrak{t}$, r, ζ and $\underline{\vartheta}$, is a unique subset of the Euclidean space $\Re^n$, the term identification, in this discussion, is not inappropriate.

The applicability of preservance, given a set of training inputs $\mathcal{T}_{\mathfrak{t}}$, to discrete sets that are not smaller than $\mathcal{T}_{\mathfrak{t}}$, suggests that generalization is not restricted merely to an extension of function specification over input regimes different (or larger) than those stated through a training set. (In § 2.2 and in the discussions of Chapter 3 and the previous sections of this Chapter generalization is interpreted as function extension.) As

[10] This is the case as the permutation operator mapping a preservance weight $\underline{w}_{<\mathfrak{t}>}$ to a (non-null) weight $\underline{w}$ is restricted to ensure $\|\underline{w}_{<\mathfrak{t}>}\| = \|\underline{w}\|$.

a consequence of generalization, the algebraic characteristics aiding a representation of the inputs, limited to the subset $\mathcal{T}_i$, are extended to a larger subset (of $\Re^n$): this extension is based on a preservance of $\mathcal{T}_\imath$ in the (internal) representation provided by the weight $\underline{w}$.

Given a training set, in particular the set $\mathcal{T}_\imath$, the translation $\underline{\vartheta}$ that all points in the discrete set ${}^{w}_{\iota}\mathcal{P}^{n}_{r}(\zeta,\underline{0})$ are to be given in order to derive the set ${}^{w}_{\iota}\mathcal{P}^{n}_{r}(\zeta,\underline{\vartheta})$ is given by the procedure indicated in Figure 4.1. The search for the translation vector $\underline{\vartheta}$ is based on the observation that all points of the basic collection of binary vectors $\mathcal{B}^n(\zeta,\underline{0})$, for all values of $\zeta \in \Re_+$, lie on a (hyper) sphere of radius $\sqrt{n}\,\zeta$. Step 2 is equivalent to the following constrained search operation:

$$\mathcal{T}_\imath(\underline{x}_j) = \max_{\zeta\in\Re_+} \mathcal{T}_\imath \cap \mathcal{B}^n(\zeta,\underline{0}),$$

for some $\underline{x}_\jmath \in \mathcal{T}_i, \jmath = 1, 2, \ldots |\mathcal{T}_\imath|$, subject to the constraint that $\mathcal{T}_\imath(\underline{x}_\jmath) \neq \emptyset$. (Note that the set of training inputs is assumed to be non-empty.) The translation $\underline{\vartheta}$ is identified, in step 3, with the centroid of the set $\mathcal{T}_i(\underline{x}_\jmath)$.

Figure 4.2 (*p.* 214) lists a procedure for determining the radix ι, scale factor ζ, and ranking index r that enable a construction of the discrete space ${}^{w}_{\iota}\mathcal{P}^{n}_{r}(\zeta,\underline{\vartheta})$ given the collection of training inputs $\mathcal{T}_\imath$: the procedure for determining the translation $\underline{\vartheta}$ is invoked to simplify the identification of the parameters of the discrete space ${}^{w}_{\iota}\mathcal{P}^{n}_{r}(\zeta,\underline{\vartheta})$. Step 2 accomplishes the centering of the training inputs. The collection of centered training inputs is denoted by $\mathcal{T}_i'$. In step 3, a partitioning is induced on $\mathcal{T}_i'$ on the basis of the norm of the vectors in the collec-

```
Given $n$, $n = 1, 2, \ldots$, and $\mathcal{T}_\imath$, a finite subset of $\Re^n$,
do
    1 Choose any member, say $\underline{x}_\jmath$, $\jmath = 1, 2, .. \; |\mathcal{T}_\imath|$, from $\mathcal{T}_\imath$.
    2 Construct the set $\mathcal{T}_\imath(\underline{x}_\jmath) \subseteq \mathcal{T}_\imath$ such that
          $\mathcal{T}_\imath(\underline{x}_\jmath) = \left\{ \underline{x} \mid \underline{x} \in \mathcal{T}_\imath \text{ and } \|\underline{x}_\jmath - \underline{x}\| = \max_{\underline{x}' \in \mathcal{T}_\imath} \|\underline{x}_\jmath - \underline{x}'\| \right\}.$
    3. Set to $\underline{\vartheta}$ a value in $\Re^n$ which ensures that $\forall \underline{x} \in \mathcal{T}_\imath(\underline{x}_\jmath)$
       $\|\underline{x} - \underline{\vartheta}\| =$ a constant.
done
```

Figure 4.1: Procedure for determining $\underline{\vartheta}$

tion, and the discrete space $\tilde{\mathcal{T}}_\imath$, $\tilde{\mathcal{T}}_\imath \subseteq \mathcal{T}_\imath'$, is constructed by choosing one (representative) input vector from each member of the partition on $\mathcal{T}_\imath'$.

The identification of the radix $\mathfrak{r}$, scale factor ζ and ranking index r are based on certain observations on the (discrete) preservance input space ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$. Recall the construction of the preservance input space (*cf*, Chapter 3). (Figure 3.5 (*p*. 134) illustrates a preservance input space of 2 dimensions and ranking index 3 with a preservance weight in $\wp_2$.) As the preservance input space is constructed by associating to each 'peripheral' input point (the $\mathfrak{r}^{nr}$ (ordered) points of the discrete set ${}_{\mathfrak{r}}\mathcal{P}^n_{r-1}(\zeta, \underline{\vartheta}) \setminus {}_{\mathfrak{r}}\mathcal{P}^n_{r-2}(\zeta, \underline{\vartheta})$, $n = 1, 2, \ldots$; $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$ in the construction described by Equation 3.7 (*p*. 172)) an n-dimensional scaled (and translated) collection of binary vectors, the specific values of the

```
Given $n$, $n = 1, 2, \ldots$, and $\mathcal{T}_\imath$, a finite subset of $\Re^n$,
do
    1 Find $\underline{\vartheta}$ appropriate to $\mathcal{T}_\imath$.
    2. Construct the finite set $\mathcal{T}_\imath^{'} \subset \Re^n$ from the set $\mathcal{T}_\imath$ with a translation of $-\underline{\vartheta}$, ie, $\mathcal{T}_\imath^{'} = \{\underline{x} - \underline{\vartheta} \mid \underline{x} \in \mathcal{T}_\imath\}$
    3. Construct the finite set $\tilde{\mathcal{T}}_\imath$ from $\mathcal{T}_\imath^{'}$ in the following manner·
       $\tilde{\mathcal{T}}_\imath = \Big\{ \underline{x}_\jmath \mid \underline{x}_1 = \arg\min_{\underline{x} \in \mathcal{T}_\imath^{'}} \|\underline{x}\|$
              and $\|\underline{x}_\jmath\| > \|\underline{x}_{\jmath-1}\|, \jmath = 2, \ldots |\mathcal{T}_\imath^{'}| \Big\}$
    4 Construct the one-dimensional function, say $f$, on $\tilde{\mathcal{T}}_\imath \setminus \{\underline{x}_1\}$ such that $f(\underline{x}_\jmath) = \|\underline{x}_\jmath\| - \|\underline{x}_{\jmath-1}\|, \jmath = 2, \ldots |\tilde{\mathcal{T}}_\imath|$.
    5. Set $\tau = 2$ if the function $f$ is unimodal, else set $\tau$ to be the same as the number of modes of $f$.
    6. Let $\Delta$ be the width of the largest cluster in $f$. Set $\zeta = \Delta(\tau^n - 1)$.
    7. Let $\delta = \min_{\underline{x} \in \tilde{\mathcal{T}}_\imath \setminus \{\underline{x}_1\}} f(\underline{x})$. Set $r = -\frac{1}{n}\frac{\log\delta}{\log\tau}$.
done
```

Figure 4.2: Procedure for determining τ, ζ and r

scale factor and the translation (governed by an exponential function of the ranking index r in a base given by the radix $\mathfrak{r}$) order the vectors in $\tilde{\mathcal{T}}_\imath$ to be separated, in the norm, in a non-uniform manner.

Figure 3.9 (*p*. 149) illustrates the nature of non-uniformity in the spacing, in the sense of the norm, of the vectors in $\tilde{\mathcal{T}}_\imath$: note that, by definition, the projection of a vector $\underline{x}$ along another vector $\underline{w}$ (from a common innerproduct space) is proportional to $\|\underline{x}\|$, the norm of $\underline{x}$. The function constructed in step 4 of the procedure in Figure 4.2 allows a relative evaluation of the intervals between adjacent projection points of the vectors in $\tilde{\mathcal{T}}_\imath$ in the linear subspace $\mathcal{L}_{\underline{w}}$ for some non-null vector $\underline{w} \in \Re^n$.

Notice that the discrete set $\tilde{\mathcal{T}}_\imath$ is constructed in such a manner as to retain only one point from every scaled collection of binary vectors whose members are found in $\mathcal{T}_\imath'$. This construction ensures that the number of modes in the function f constructed in step 4 is no more than the smallest radix $\mathfrak{r}$ necessary in describing the vectors in the collection $\mathcal{T}_\imath$. (Note that translation, scaling and rotation do not affect the radix of numbering.) More accurately, f is unimodal if the members of $\mathcal{T}_i$ are derived from scaled and translated collections of binary vectors. For radices that are higher than 2, the number of modes in f is identical to the radix. This aspect is incorporated in step 5.

Proposition 3.1.9 (*p*. 136) shows that the discrete space $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ contains points of $\Re^n$ sampled from open balls of finite radius and centered at the vertices of $\mathcal{B}^n(\zeta, \underline{\vartheta})$. This property is also true of preservance input spaces whose radix is higher that 2. The function f constructed on the discrete set $\tilde{\mathcal{T}}_\iota$ essentially clusters the vectors around the modal values as the value of f shrinks exponentially around each modal value. Step 6 sets the scale factor ζ to accommodate the widest cluster observed in $\tilde{\mathcal{T}}_\iota$. In step 6, the ranking index is assigned a value which will allow the smallest interval between adjacent points in $\tilde{\mathcal{T}}_\iota$ (adjacency is in the sense of the norm) to be recreated in the preservance input space ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$.

Figure 4.3 (*p.* 217) lists a procedure for determining a weight $\underline{w}$ that will be a preservance weight for the given collection of inputs in the training set. The basis for this procedure is that the discrete space ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta, \underline{0})$ is a rotated version of ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta, \underline{0})$. The constructions in the procedure are aimed at facilitating an evaluation of approximation only over vectors of equal norms. I have assumed that the ranking index r is adequate enough to allow every input vector in the training set to be approximated by a vector in the (centered) preservance input space with no error in the norm: a relaxation of this assumption will necessitate a revision of the construction of ${}^{\underline{w}}{}_k\mathcal{P}(\xi)$ in step 5 to let the argument ξ of the set ${}^{\underline{w}}{}_k\mathcal{P}(\xi)$ take on values in (connected) intervals of the set ${}^{\underline{w}}{}_k\tilde{\mathcal{P}}$.

Approximation of vectors in $\mathcal{T}_\iota$, the collection of training input points, by the vectors of the preservance input space is governed by an error

```
Given $n$, $n = 1, 2, \quad , \mathcal{T}_\imath$, a finite subset of $\Re^n$ and $\epsilon > 0$
do
    1. Find $\mathfrak{t}$, $\zeta$ and $r$ appropriate to $\mathcal{T}_\imath$.
    2. Construct the discrete sets $\mathcal{T}_\imath^{'}$ and $\tilde{\mathcal{T}}_\imath$ as in steps 2 and 3, respectively, of the procedure in Figure 4 2.
    3. Partition the set $\mathcal{T}_\imath^{'}$, on the basis of the norm, to derive the sets
       $$\mathcal{T}_\imath^{'}(\xi) = \left\{ \underline{x} \mid \underline{x} \in \mathcal{T}_\imath^{'},\ \|\underline{x}\| = \xi \right\} \ \forall \xi \in \tilde{\mathcal{T}}_\imath$$
    4. Set $k = 1$. Let $\underline{w}_k \in \Re^n \setminus \{\underline{0}\}$ be the current estimate of $\underline{w}$.
    5. Construct the following sets: $\frac{\underline{w}_k}{\mathfrak{t}}\mathcal{P}_r^n(\zeta, \underline{0})$,
       $${}^{\underline{w}_k}\bar{\mathcal{P}} = \left\{ \underline{x}_j \in \frac{\underline{w}_k}{\mathfrak{t}}\mathcal{P}_r^n(\zeta, \underline{0}) \mid \underline{x}_1 = \underset{\underline{x} \in \frac{\underline{w}_k}{\mathfrak{t}}\mathcal{P}_r^n(\zeta, \underline{0})}{arg\ min} \|\underline{x}\| \right.$$
       $$\left. \text{and } \|\underline{x}_j\| > \|\underline{x}_{j-1}\|, j = 2, \quad . \left| \frac{\underline{w}_k}{\mathfrak{t}}\mathcal{P}_r^n(\zeta, \underline{0}) \right| \right\}$$
       and ${}^{\underline{w}_k}\mathcal{P}(\xi) = \left\{ \underline{x} \mid \underline{x} \in \frac{\underline{w}_k}{\mathfrak{t}}\mathcal{P}_r^n,\ \|\underline{x}\| = \xi \right\} \ \forall \xi \in {}^{\underline{w}_k}\bar{\mathcal{P}}$.
    6 Evaluate $e = \sum_{\xi \in \mathcal{T}_\imath^{'}(\xi) \cap {}^{\underline{w}_k}\mathcal{P}(\xi)} |\mathcal{T}_\imath^{'}(\xi) - {}^{\underline{w}_k}\mathcal{P}(\xi)|$
    7. If $e > \epsilon$ find a new guess $\underline{w}_{k+1}$ based on $\underline{w}_k$ and $e$, increment $k$ and repeat steps 5 and 6.
    8. Set $\underline{w}$ to the current estimate.
done
```

Figure 4.3: Procedure for determining $\underline{w}$

criterion stating the mismatch (in terms of an appropriately chosen metric) of the collection of vectors of similar (identical) norms, ie, the error criterion measures the mismatch (or unsatisficability) of $\mathcal{T}_1'(\xi)$ with respect to $^{\underline{w}}\mathcal{P}(\xi)$ for all values of ξ that occur in $\tilde{\mathcal{T}}_1$ as well as $^{\underline{w}_k}\tilde{\mathcal{P}}$, for all values of k, $k = 1, 2,$ The distance between two discrete sets $\mathcal{A}_1$ and $\mathcal{A}_2$ is defined as in the following:

$$|\mathcal{A}_1 - \mathcal{A}_2| = \max_{x \in \mathcal{A}_1} \min_{y \in \mathcal{A}_2} |x - y|$$

The procedures listed in this section give an insight into the issue of identification of preservance input spaces given a collection of training inputs. As indicated in an earlier section, the identification of a preservance input space given a collection of training inputs also, uniquely, identifies the class of preservance weights associated with the inputs in the training set. The choice of weights in the distinct nodes of the (first layer of a layered) neural signal processor, on an identification of a preservance input space, reduces to an enumeration within the class of preservance weights. Note that the identification of preservance input spaces is related only to the problem of representing the input signal space and is not influenced by the kind of mapping realized.

4.4 Symbolic Computation with Neural Networks

The representation of functions (processors) in neural networks wherein the first (input) layer incorporates preservance weights is essentially a realization of mappings on discrete input spaces as discussed in the preceding sections. It is not difficult to visualize a situation, typically when the output takes on discrete values, wherein the isolated neurons and the layered networks involving such neurons compute functions between symbol spaces.

Considerations of computability necessitate the symbol spaces to be discrete. The task of incorporating discrimination on symbol spaces, then, reduces to an identification of a suitable mapping between the discrete spaces employed by neural signal processors and the symbol spaces that support the function to be realized through the neural signal processor. In neural networks, inter-processor interconnection strengths are commonly identified with the mechanism of representing available (*ie* given) knowledge about the functional association between inputs and outputs. Emphasis is laid on the aspect that the representation is not specific to any processor in the ensemble. In the tradition of symbolic computation, on the other hand, the notion of strengths of interconnection between processing nodes is not commonly used.

Parameterization and the consequent representational methodology is always considered internal to processors with the implication that representation of available knowledge is processor specific. This necessitates a reinterpretation of isolated neurons, for the purposes of discussing the possibility of symbolic computation with neural signal processors, as incorporating, in its definition, the weights of the channels incident on the processor and considering the representation incorporated by these weights as being specific to that processor; however, this reinterpretation will be restricted only to this section

Symbolic computation is studied primarily through the frameworks of Turing Machines, Cellular Automata *etc.* In a schema of interconnected processors, the basic processor is formalized as a (rewriting) rule specifying the value assigned to the output given a configuration of input values. Representation of functions in neural signal processors is through an assignment of appropriate values to weights and thresholds. Consequently, a study relating the values assigned to weights and thresholds to the structure of functions realized by neural signal processors is essential in understanding the nature of symbolic computation offered by neural signal processors.

Isolated neurons, in the context of Boolean functions, represent linearly separable dichotomies and the representation of Boolean functions, in general, needs at least two layers of discrimination as established by **Lippmann**, 1987 and others. Symbolic computation achievable by neural signal processors, thereby, reduces to a study of the equivalent of linear separability in the context of symbol spaces.

Symbolic spaces have only structural attributes. Thus, the discussion has to proceed through an algebraic characterization of the discrete input space ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^{n}_{r}(\zeta, \underline{\vartheta})$, $\underline{w} \in \Re^n$, $\|\underline{w}\| \neq 0$, $\mathfrak{r} = 1, 2, \ldots; n = 1, 2, \ldots; \zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$. Without any loss of generality, the subsequent discussion will be based on the discrete input space ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ with weights $\underline{w} \in {}_{\mathfrak{r}}\wp_n(\alpha)$, for any appropriate $\alpha \in \Re_+$, noting that to every weight $\underline{w} \in \Re^n$ such that $\|\underline{w}\| \neq 0$ there exists a preservance input space ${}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ isomorphic to ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ for all admissible values of $\mathfrak{r}$, n, ζ and $\underline{\vartheta}$ and that ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, for all admissible ζ and $\underline{\vartheta}$, is a preservance input space for all weights in ${}_{\mathfrak{r}}\wp_n(\alpha)$ as indicated in § 3.4 (*p.* 171).

A function, by definition, induces a partition on the relevant input space and, hence, a characterization of the members of the partition is equivalent to a characterization of the function: this equivalence will be considered, where necessary, while relating the weight and threshold values to the algebraic structure of the functions realized by neural signal processors. The partition induced on a set, say $\mathcal{A}$, will be denoted by $\mathfrak{P}(\mathcal{A})$, the distinct members being denoted by $\mathfrak{P}_i(\mathcal{A})$, $i = 1, 2, \,..|\mathfrak{P}(\mathcal{A})|$:

$$\begin{aligned}\mathfrak{P}(\mathcal{A}) &= \{\mathfrak{P}_1(\mathcal{A}), \mathfrak{P}_2(\mathcal{A}), \ldots \mathfrak{P}_j(\mathcal{A})\}, \text{ for some } j = 1, 2, \ldots, \\ \mathcal{A} &= \bigcup_{i=1}^{|\mathfrak{P}(\mathcal{A})|} \mathfrak{P}_i(\mathcal{A})\,;\, \mathfrak{P}_i(\mathcal{A}) \cap \mathfrak{P}_j(\mathcal{A}) = \emptyset \;\forall i, j = 1, 2, \ldots |\mathfrak{P}(\mathcal{A})|\,,\, i \neq j\end{aligned}$$

Neural signal processors will in this discussion be considered as conforming to a processing model of two layered neural networks, *ie*,

$$y(\underline{x}) = \sigma(\sum_{i=1}^{n} v_i y_i(\underline{x}) - \theta)\,,\, \forall \underline{x} \in \Re^n, \tag{4 5}$$

where, $y_i(\underline{x}) = \sigma(\underline{w}_i\ \underline{x} - \theta_i)$ and the other terms have the interpretation indicated earlier in this chapter. The discrete space $\mathcal{B}^n$, $n = 1, 2, \ldots$, together with a partial ordering relation, denoted by $\preceq$,[11] and binary

[11] A typical example is 'less than or equal to'.

operations $\vee$ and $\wedge$ which have the interpretations of supremum and infimum, respectively, is termed a lattice: the operations $\vee$ and $\wedge$ are assumed to be closed over $\mathcal{B}^n$.

It is not difficult to visualize that the lattice property of $\mathcal{B}^n$ is unaffected by scaling and/or translation. However, the possibility of a necessity for redefining the operations $\vee$ and $\wedge$ under a scale factor $\zeta \in \Re_+$ and a translation $\underline{\vartheta} \in \Re^n$ is supported by the notation $\vee_{(\zeta,\underline{\vartheta})}$ and $\wedge_{(\zeta,\underline{\vartheta})}$ respectively. The following is easily established.

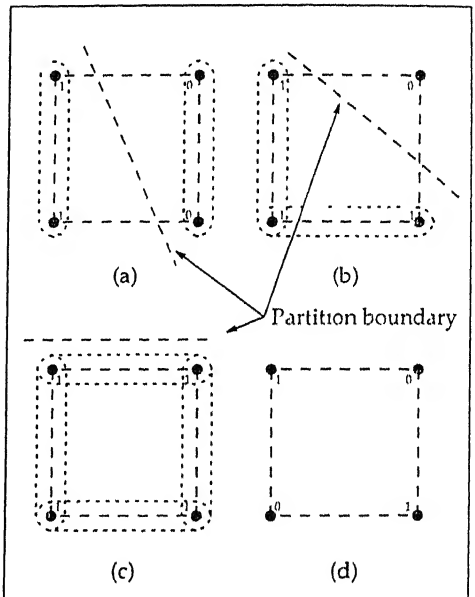


Figure 4.4: Partitions on $\mathcal{B}^n$

PROPOSITION 4.4.1 *The members in a partition of any linearly separable dichotomy on $\mathcal{B}^n$ are expressed as*

$$\mathfrak{P}_\imath(\mathcal{B}^n) = \bigcup_{\jmath_\imath=1}^{k_\imath} \mathcal{B}_{\imath_\jmath},$$

where $\imath = 1, 2$; $k_i = 0, 1, \ldots,$ $k_1 + k_2 \neq 0$, and each set $\mathcal{B}_{\imath_\jmath}$ is isomorphic to $\mathcal{B}$ and aligned 'parallel' along any one of the co-ordinate axes.

This statement is an immediate consequence of the notion of Cartesian products and the geometric notion of convexity that each member of the partition in a linearly separable dichotomy is subjected to. (See Chapter 2 for the geometric notion of linear separability.)

Figure 4.4 illustrates some instances of partitions induced on $\mathcal{B}^2$: cases (a) through (c) depict linearly separable dichotomies and case (d) depicts a dichotomy that is not linearly separable. (In this illustration, the binaries that form a one-dimensional space isomorphic to $\mathcal{B}$ are marked by ovals.) The illustration implies the following without the requirement of a proof.

PROPOSITION **4.4.2** *For every dichotomy on $\mathcal{B}^n$, $n = 1, 2,$, that is not linearly separable infimum and/or supremum operations are not satisfied by every pair of points in at least one member of the partition.*

The above two statements motivate the following.

THEOREM **4.4.1** *The following are equivalent.*

1. *A dichotomy on the discrete space $\mathcal{B}^n$ is linearly separable.*
2. *Every member of the partition induced by a dichotomy on $\mathcal{B}^n$ is a semi-lattice.*

Before providing a proof for this statement, it is worthwhile to note that a set, say $\mathcal{A}$, is a semi-lattice provided the following axioms are satisfied.

A1. The partial ordering relation $\preceq$ is closed over $\mathcal{A}$.

A2 At least one of the binary operations $\vee$ and $\wedge$ is closed over $\mathcal{A}$

Proof: The proof will be given in four stages based on the different kinds of partitions induced by linear separable functions as illustrated in Figure 4.4 (*p*. 222). Though the illustration shows partitions on $\mathcal{B}^n$, the types of partitions and the arguments are sufficiently general to be applicable to partitions on $\mathcal{B}^n$ for values of n other than 2.

Case 1 Trivial Functions One member of the partition is null

See (c) of Figure 4.4. In this case $\mathfrak{P}(\mathcal{B}^n) = \{\emptyset, \mathcal{B}^n\}$. Both the members of the partition are lattices and the statement is true.

Case 2 Non-trivial Functions· One member of the partition is a singleton

See (b) of Figure 4.4. The partition in this case is expressed, in general, as $\mathfrak{P}(\mathcal{B}^n) = \{\{\underline{x}_0\}, \mathcal{B}^n \setminus \{\underline{x}_0\}\}$, where $\underline{x}_0 \in \mathcal{B}^n$. It is immediately apparent that the singleton set $\{\underline{x}_0\}$ is a lattice, and thereby a semi-lattice, as all the axioms (of a lattice) are trivially satisfied. Considering the other member of the partition, the geometrical equivalent of the notion of linear separability implies that $\{\underline{x}_0\}$ is either the maximal or the minimal element of $\mathcal{B}^n$, *ie*,

$$\forall \underline{x} \in \mathcal{B}^n \ \underline{x}_0 = \vee(\underline{x}, \underline{x}_0) \text{ or } \underline{x}_0 = \wedge(\underline{x}_0, \underline{x}) .$$

This assures that $\mathcal{B}^n \setminus \{\underline{x}_0\}$ is a semi-lattice.

Case 3. Non-trivial Functions: Partition members are neither null nor singletons

See (a) of Figure 4.4. Each member of the partition is expressed as a union of one-dimensional spaces, each isomorphic to $\mathcal{B}$, *ie*,

$$\mathfrak{P}_i(\mathcal{A}) = \bigcup_{j=1}^{k_i} \mathcal{B}_{i_j},\, i = 1, 2,$$

where, $\mathcal{B}_{i_j}, j = 1, 2, \;.\; k_i$, is isomorphic to $\mathcal{B}$ for some $k_i = 1, 2, \;.\;\;.\;$, such that for every $\mathcal{B}_{i_{j_1}} \subset \mathfrak{P}_i$ there exists $\mathcal{B}_{i_{j_2}} \subset \mathfrak{P}_i$, $j_2 \neq j_1$ and $\mathcal{B}_{i_{j_1}} \cap \mathcal{B}_{i_{j_1}} \neq \emptyset$. This aspect ensures that

$$\mathcal{B}_{i_{j_1}} \cap \mathcal{B}_{i_{j_2}} \neq \emptyset \Rightarrow \underline{x}_1 = \vee(\underline{x}_2, \underline{x}_3) \text{ or } \underline{x}_1 = \wedge(\underline{x}_2, \underline{x}_3),$$

$$\{\underline{x}_1\} \equiv \mathcal{B}_{i_{j_1}} \cap \mathcal{B}_{i_{j_2}}, \; \{\underline{x}_2, \underline{x}_3\} \equiv \mathcal{B}_{i_{j_1}} \Delta\, \mathcal{B}_{i_{j_2}}$$

Note the structure of the members of the partition. While the operation $\vee$ is defined over all pairs of points in one member, say $\mathfrak{P}_i(\mathcal{A})$, this operation is not defined over all pairs of points in the complementary member $\mathfrak{P}_{\overline{2-i}}(\mathcal{A})$. In a similar way the operation $\wedge$ is defined over all pairs of points in $\mathfrak{P}_{\overline{2-i}}(\mathcal{A})$ and is not defined over all pairs of points in $\mathfrak{P}_i(\mathcal{A})$. A simple justification for this situation lies in the fact that under the binary operations $\vee$ and $\wedge$, while the (global) supremum belongs to $\mathfrak{P}_i(\mathcal{A})$, for some $i = 1, 2$, the global infimum belongs to the complementary member $\mathfrak{P}_{\overline{2-i}}(\mathcal{A})$.

Case 4. Functions not linearly separable

See (d) of Figure 4.4. From Proposition 4.4.2 (p. 223) it is clear that none of the members of the partition satisfies the axioms of a semi-lattice.

□

The discrete space $\mathcal{P}^n_r(\zeta,\underline{\vartheta}) \equiv {}_2\mathcal{P}^n_r(\zeta,\underline{\vartheta}), n = 1,2,\ .; r = 1,2,..\ ; \zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$, being derived from $\mathcal{B}^n$ through scaling and translation, Theorem 4.4.1 (*p.* 223) is easily extended to the case of dichotomies over $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ as the following.

Theorem **4.4.2** *Every member of the (binary) partition induced by a linear separable dichotomy on $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$, $n = 1,2,.\ .; r = 1,2,\quad ; \zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$, is a semi-lattice.*

If the radix $\mathfrak{r}$ is considered not to be restricted to 2, the following is easily established.

Theorem **4.4.3** *All members in a (binary) partition induced by a linearly separable dichotomy on ${}_{\mathfrak{r}}\mathcal{H}^n(\zeta,\underline{\vartheta})$, $\mathfrak{r} = 1,2,..\ ; n = 1,2,...;$ $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$, is a semi-lattice.*

It is worthwhile to note that ${}_{\mathfrak{r}}\mathcal{H}^n(\zeta,\underline{\vartheta})$ for all the admissible values of $\mathfrak{r}$, n, ζ and $\underline{\vartheta}$, forms a (truncated) lattice points (**Erdös, Gruber & Hammer**, 1989) and hence is a semi-lattice. Here again the partitions are of the same four kinds as discussed in the proof of Theorem 4.4.1.

Theorem **4.4.4** *The following statements are equivalent*

1. *A bipolar dichotomy $\mathfrak{D}$: ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta}) \to \mathcal{B}$ is linearly separable.*
2. *Every member of $\mathfrak{P}({}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta}))$ is a semi-lattice.*

Theorem 4.4.4 (*p.* 226), which captures the essence of the algebraic characterization of linear separable dichotomies on discrete spaces, is effectively a definition of linear separability for mappings defined on symbolic spaces. (Note that the notion of linear separability, as considered presently in the literature, makes sense only for dichotomies.) For the sake of completeness, the following definition of linear separability in connection with symbolic spaces is being provided.

DEFINITION **4.4.1** *A dichotomy over a symbol space, itself embedded in a (semi) lattice, is linearly separable if each component in the partition induced by the dichotomy on the symbol space is a sub semi-lattice.*

4.5 Summary

Single layered neural signal processors, the simplest non-trivial interconnected ensemble of neurons, in a continuation of the discussion of preservation of discrete spaces in one-dimensional spaces, have been shown to be adequate in representing all dichotomies and functions of interest on discrete spaces. While the adequacy of single layer neural processing structures and the associated exponential dependence of the number of processing nodes on the input space dimensionality have long been known in literature, a discussion based on the notion of preservance weights and preservance input spaces, as carried out in this chapter, has shed some light on the algebraic characteristics resulting from a restriction of the input space.

The number of processing nodes required to achieve the desired representation, in general, shows an exponential dependence on the cardinality of the discrete space under consideration, and given the construction of discrete spaces ${}^{\underline{w}}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$ the number of processing nodes becomes undesirably large as a function of the input space dimensionality n as well as r. The cardinality of this discrete input space is unaffected by the specific choices of non-null weight $\underline{w} \in \Re^n$, $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$ and it is to be expected that the number of processing nodes is invariant to these parameters.

Learning (with generalization) of the weights, thresholds, bias (θ) and co-efficients of linear combination ($\underline{v}$) has been shown to relate to the simple procedure of learning in isolated neurons; the additional requirement of specifying $\underline{v}$ is accomplished through a solution to a linear system of equations. Motivated by the urge to seek layered neural information processing structures capable of function representation with complexities less demanding than that of single layer neural signal processors, representation in multi layered neural signal processors has been investigated. While these network structures donot enlarge the space of functions represented relative to that seen in single layered neural signal processors, realization of given dichotomies (functions) is accommodated with fewer number of processing nodes.

Function realization, or processor representation in neural networks, is interpreted to mean an identification of a suitable preservance input space given examples of processing followed by an approximation of the desired processing functionality as functions on the preservance input space Identification of a preservance input space given a collection of training input also uniquely identifies the class of preservance weights. The issue of choosing weights in the distinct nodes of the first layer, then, reduces to an enumeration in the class of preservance weights. Representation of the input signal space is accomplished through an identification of a preservance input space given a training set. This implies that learning or representation in neural networks is not the same as an incorporation of values in a look-up table (or memory) as suggested by **Aleksander** (1983b) and **Stonham** (1983).

Neural information processing having been discussed in the previous and the present chapters on discrete spaces, a natural curiosity of seeking the possibility of realizing symbolic computation through layered neural information processing structures has motivated a study of neural signal processors in terms of the partitions induced on the (discrete) input space. An inquiry into the algebraic properties of the partitions leads to the conclusion that linear separability of a given dichotomy is equivalent to the members of the partitions induced by the given dichotomy being semi-lattices. This alternative definition of linear separability allows the notion of isolated neurons and, consequently, neural information processing structures to be defined on symbol spaceš, however, in order that the notion of learning and, in particular, generalization are valid on symbol spaces, an equivalent of linear combination, as an operation on symbols, is imperative.

The algebraic characterization of linear separability, together with the notion of neural signal processing architectures minimal to a given dichotomy prompt an inquiry, not unnaturally, into the nature of the representational paradigm operative in neural information processing. In particular, it is of interest to be able to identify an axiomatic framework that would provide an insight into the kinds of processing that can be expected of neural signal processors. Noting that representation of functions in neural signal processing structures involving a degree of layering larger than unity entail a reduction in the requirement of processing nodes, the character of representation, in the sense of the nature and degree of approximation, of layered neural signal processors needs to be investigated. Chapters 5 and 6 will be devoted to these and related inquiries.

Chapter 5

Neural Signal Processing Architectures: Representational issues

> Marco Polo describes a bndge, stone by stone.
>
> 'But which is the stone that supports the bridge?' Kublai Khan asks.
>
> 'The bridge is not supported by one stone or another,' Marco answers, 'but by the line of the arch that they form.'
>
> Kublai Khan remains silent, reflecting. Then he adds: 'Why do you speak of the stones? It is only the arch that matters to me.'
>
> Polo answers: 'Without stones there is no arch.'
>
> — Italo Calvino
> in *Invisible Cities*,
> Picador, London, 1979.

Signal processor realization, especially of the nonlinear kind, through artificial neural networks is centered in the *model free* classification/ estimation aspect of the paradigm which essentially simplifies the operational nature and allows for a representation of input-output dependencies to be established through an appropriate number of non-exhaustive examples. Processor representation has been looked upon in neural networks, as in signal processing, as a situation of approximating a function, however, the deviation from conventional signal processing approaches lies in that the function approximated is expected to have connotations of perceptual characteristics, specifically categorization/estimation, and the function is commonly described, not analytically, but in terms of a list, generally not exhaustive, of typical correspondences between inputs and outputs· the study of representation in Chapters 3 and 4 are based on these aspects.

Function approximation in neural networks, similar to that in conventional signal processing, is viewed as a search, for the closest member (in the sense of an appropriate measure of satisfiability[1]), in the linear span of an appropriately chosen set of 'basis functions,' thereby suggesting close relationships between function realization in neural networks and integral transforms. The 'basis functions,' though dependent on the specific class of functions being approximated (as decided by the particular repertoire of input-output associations provided in the training set), are at a conceptual level, particularly in layered neural

[1] Refer § 2 1 for the notion of satisfiability employed in search.

networks, synthesized from a family of activation functions which is, in general, shared by a large class of training data: the family of activation functions is independent of the family of functions to be approximated.

In neural signal processing, the specific choice of 'basis functions' is determined from the available knowledge of the desired map by a procedure of parameter location, also known as learning, or training. Clustering, in the abstract sense of capturing the relevant features needed for categorization, estimation, or function approximation, is the essence of realizing desired maps whereby the 'basis functions' are constrained to be nonlinear and to have localized influence, not necessarily in the sense of having compact supports. The activation functions (σ) in the processing nodes, too, are, thereby, required to exhibit a nonlinear functional nature: the kind (or type) of nonlinearity is crucial to the discriminatory power of a neural signal processor.

Monotonicity in the activation functions, components in the synthesis of basis functions (of local influence), is not considered favorable in function approximation in view of the complexity in generating local functions as a linear combination of non-local functions, essentially an algorithmic (convergence) consideration. However, these non-local functions, sigmoidal functions being a typical example, are simple, adequate for function approximation and considered to be biologically plausible formalisms of the discriminatory requirement in decision making. The activation functions, though monotonic, have local variations.

While the representation potential and characterization of the existence of representation, in neural signal processors with a single layer of decision making has been studied in sufficient detail, not enough attention has been riveted on the capability of multi-layered neural signal processors, not withstanding the algorithmic inadequacies of such processor realization schemes. Partly hindered, in analytical treatment, by the intervening nonlinearities between (adjacent) layers of linear filtering, the role of multiple layers of decision making, particularly with bounded number of processing nodes, on function approximation capabilities is not known in desired detail. An immediate consequence of this inadequacy is that no satisfactory criterion is yet available for deciding the number of layers[2] to be used in a neural signal processor, a requirement of processor design in the neural processing paradigm

I will initiate the discussion of representational issues in neural signal processing by trying to provide an understanding of the underlying (operational) paradigm in artificial neural networks. This exercise is important as neural networks, in the literature, have been compared at a functional level with approaches based on Gabor functions (**Daugmann**, 1988), ridge functions (**Ya Lin & Pinkus**, 1993), wavelets (**Zhang & Benveniste**, 1991; **Pati & Krishnaprasad**, 1993),

[2] While this statement is applicable, largely, to neural signal processors realized through activation functions of the sigmoidal (including hard limiter) type, and it is unlikely that multiple layers of decision making would be needed, for satisfactory function approximation, with radial (or elliptical) basis functions, knowledge of the role of multiple layers of decision making would be useful, in general, in explorations of processor types capable of capturing our (current) understanding of perceptual abilities and requirements.

and matched filtering (**Grant & Sage**, 1986), and have been applied in many signal processing situations, *eg*, image data compression, routing and congestion control of telecom traffic, decoding in code-division multiple access (**Aazhang**, **Paris & Orsak**, 1992), *etc*. Moreover, the superiority, in modeling of neural networks to conventional approaches has often been reported in the literature.

Neural signal processors are shown in this chapter to provides statements of aggregations of decisions taken on features extracted from patterns presented to the network: the features are related to patterns through integral transforms and the averaging process allows concepts to be aggregated from decisions on relevant features. This aspect of the underlying paradigm of artificial neural networks enables a discovery of similarities and dissimilarities with conventional signal processing approaches and assures a possibility of a complete neural basis for realization of nearly all aspects of a signal processor, the latter assurance can, however, be given only when a multi-layered scheme is adopted for the approximation task at hand. Cognitive scientists and neuro-anatomists could find this perspective of neural networks incorporating feature extraction, decisions on feature spaces, and aggregation of decisions to form concepts, useful in exploring the specific kinds of signal processing that are carried out in the nervous system.

A study of the representational potential offered by multi-layered neural signal processors being the main topic of this chapter, I begin with the definition of neural signal processors in § 5 1 and establish that these neural signal processors are capable of representing continuous functions on finite dimensional spaces with arbitrary accuracy: neural signal processors, though introduced in Chapter 4, are defined once again to enable an investigation into the kinds of signal processing situations represented. The functional character of neural signal processors are studied in § 5.2 (*p*. 249) and in this section the axioms of neural signal processing are formulated.

In § 5.3 (*p*. 276) I formulate the representational paradigm operative in neural signal processors as being a nonlinear association between integral transforms: this paradigm introduces the metaphor of integral transform kernels being the internal representation of the knowledge available (or given) through a training set. § 5.4 (*p*. 284) is a study of representation in neural signal processors, in particular, the character of representation, wherein I suggest an interpretation to a function representation theorem of Kolmogorov in the context of neural signal processing. I also establish that as the depth of layering increases, the degree of smoothness with which a multi-variate function is realized increases: this in turn suggests an explanation for the relatively higher representational complexity of single layer neural signal processors as compared to the multi-layered variety.

5.1 Neural Signal Processors: Definition and representational potential

In § 2.2, the formal models for neurons and neural networks have been introduced and in Chapter 4 the functional model of layered neural signal processors has been considered. These models have been suggested to address issues of information storage (*ie*, representation) and processing that dominate any discussion of automated intelligence viewed in the perspective of information processing. Neural signal processors are formulated as linear combinations of neural responses to overcome the inevitable restriction of the output in the formal model of neurons (purporting to capture the firing frequency of the action potential of biological neurons) being limited to a specific (proper) subset of $\Re$, the real number field, typical examples being closed intervals like $[0, 1]$ or $[-1, 1]$ in the case of continuous real valued neurons, and sets like $\{0, 1\}$ or $\{-1, 1\}$ with binary (real) valued neurons.

Multi-layered neural networks form the basis of neural signal processors and the possibility of representing functions in feed-forward multi-layered neural signal processors with fewer processing nodes than necessary in the case of single layer neural signal processors encourages a consideration of more general schemes of neural processing. Noting that the model of processing in networks of neurons suggested by Equation 2.16 (*p*. 67) is a unified statement of neural network architectures, the following is suggested.

𝔇𝔈𝔉𝔍𝔑𝔍𝔗𝔍𝔒𝔑 **5.1.1** *The class of scalar neural signal processors of type-k, $k = 1, 2, .\ \ $, denoted by ${}^{k}\mathfrak{N}$, is given by the functional form*

$$\begin{aligned}
\dot{\eta}^{(1)}_{j^{(1)}}(\underline{x}, t) &= a^{(1)}_{j^{(1)}}(\eta^{(1)}_{j^{(1)}}(\underline{x}, t)) \Bigg[b^{(1)}_{j^{(1)}}(\eta^{(1)}_{j^{(1)}}(\underline{x}, t)) + \underline{s}^{(1)}_{j^{(1)}}\, \underline{x} \\
&\qquad + \sum_{i=1}^{m_{j^{(1)}}} \epsilon^{(1)}_{j^{(1)}\,i}\, y^{(1)}_{i}(\underline{x}, t - \tau^{(1)}_{r_{j^{(1)}\,i}}) \Bigg], \\
y^{(1)}_{j^{(1)}}(\underline{x}, t) &= \sigma^{(1)}_{j^{(1)}}(\eta^{(1)}_{j^{(1)}}(\underline{x}, t)), \quad j^{(1)} = 1, 2, \quad m_1, \\
&\qquad \textit{for some } m_1 = 1, 2, \quad ., \qquad\qquad (5.1a) \\
\dot{\eta}^{(\ell)}_{j^{(\ell)}}(\underline{x}, t) &= a^{(\ell)}_{j^{(\ell)}}(\eta^{(\ell)}_{j^{(\ell)}}(\underline{x}, t)) \Bigg[b^{(\ell)}_{j^{(\ell)}}(\eta^{(\ell)}_{j^{(\ell)}}(\underline{x}, t)) \\
&\qquad + \sum_{i_f=1}^{m_{j^{(\ell-1)}}} s^{(\ell)}_{j^{(\ell)}\,i_f}\, y^{(\ell-1)}_{i_f}(\underline{x}, t - \tau^{(\ell)}_{f_{j^{(\ell)}\,i_f}}) \\
&\qquad + \sum_{i_r=1}^{m_{j^{(\ell)}}} \epsilon^{(\ell)}_{j^{(\ell)}\,i_r}\, y^{(\ell)}_{i_r}(\underline{x}, t - \tau^{(\ell)}_{r_{j^{(\ell)}\,i_r}}) \Bigg], \\
y^{(\ell)}_{j^{(\ell)}}(\underline{x}, t) &= \sigma^{(\ell)}_{j^{(\ell)}}(\eta^{(\ell)}_{j^{(\ell)}}(\underline{x}, t)), \\
&\qquad j^{(\ell)} = 1, 2, \ldots m_\ell, \quad \textit{for some} \quad m_\ell = 1, 2, \ldots, \\
&\qquad \ell = 2, 3, \ldots k, \qquad\qquad (5.1b) \\
\mathfrak{y}^{(k)}(\underline{x}, t) &= \sum_{j=1}^{m_k} w^{(k+1)}_{j} y^{(k)}_{j}(\underline{x}, t) - \theta^{(k+1)}, \qquad\qquad (5.1c)
\end{aligned}$$

where η, y, $\underline{x}$, t, σ, a, b, s, ϵ, and τ have the same connotations[3] *as indicated in* § 2.2. *The processor, and its functionality, are denoted by $\mathfrak{y}^{(k)}$ with appropriate indices and dependencies.*[4]

[3] Note that thresholding is incorporated in the abstract translation function b

[4] Note that the neural computational process defined as a transformation on an m_0

(Note that this definition is the same as the functional form in Equation 2.16 (*p.* 67) except for the additional statement expressing the mechanism of deriving the required scalar output $\mathfrak{y}^{(k)}$ from the array of decisions $\underline{y}^{(k)}$ and subsumes the definition of neural signal processors introduced in Chapter 4.)

The above definition can be effortlessly extended to the class of vector neural signal processors by the assignment

$$\underline{\mathfrak{y}}^{(k)}(\underline{x}, t) \triangleq \underline{\eta}^{(k+1)}(\underline{x}, t) \tag{5.2}$$

(where, the elements of $\underline{\eta}^{(k+1)}$ is defined as in Equation 5.1), though in signal processing, often, the processor is scalar valued, and, unless otherwise mentioned, neural signal processors will be considered to be scalar valued in the sequel. In passing, it is worthwhile to note that ${}^k\mathfrak{N}$ is a function[5] space described by

$$ {}^k\mathfrak{N} = \bigcup_{\underline{m}^{(k)}} {}^k\mathfrak{N}_{m_0,m_1, \quad m_k,m_{k+1}}, \tag{5.3}$$

where, ${}^k\mathfrak{N}_{\underline{m}^{(k)}}$ ($\triangleq {}^k\mathfrak{N}_{m_0,m_1, \quad m_k,m_{k+1}}$ with $\underline{m}^{(k)} = [m_0, m_1, \ldots m_k, m_{k+1}]^{T}$) denotes the family of functions realized by a type-k neural signal processor with the number of nodes in the corresponding layers[6] given by the elements of the vector $\underline{m}^{(k)}$.

dimensional pattern space, and indexed in a (directed) one-dimensional space signified to have the interpretations of time, can easily be extended as a neural computational field wherein the indexing space is of dimensionality greater than one However, specific relations of (partial) ordering need to be imposed.

[5] More precisely, as indicated later, ${}^k\mathfrak{N}$ is an operator space

[6] m_0 denotes the number of inputs, *ie*, the number of elements in $\underline{x}$, and $m_{k+1} = 1$ in the case of scalar neural signal processors

Figure 5.1 illustrates the processing hierarchy in a neural signal processor of type-k. Neural signal processors, of all types, have a taxonomy, and architectural peculiarities, similar to neural networks[7] in view of the above definition. Thus, neural signal processors can be continuous or discrete (generally binary) valued, feed-forward, recurrent or competitive, with additive or multiplicative (shunting) dynamics in the individual processing nodes. In the above definition, the reasons for associating the number of decision-making layers, *ie*, k, with the type number of neural signal processors and the necessity for the notion of types will become clear in the ensuing discussion.

Present research on neural network based signal processing and function approximation has focused extensively on type-1 neural signal processors (*ie*, ${}^{1}\mathfrak{N}$). According to a theorem due to **Cybenko** (1989), type-1 neural signal processors with sigmoidal activation functions realize continuous functions with arbitrary accuracy. This result has also been reported by several others in the literature (see *eg*, **Vepsäläinen**, 1991; **Mhaskar**, 1993; **Ya Lin & Pinkus**, 1993). We note, in passing, that an isolated neuron is expressible as a type-k neural signal processor for all values of k ($k = 1, 2, \ldots$), *ie*, isolated neurons belong to the space of processors realized from them.

On being introduced to the definition of neural signal processors, one of the first questions that crops up concerns the potential for rep-

[7] See Chapter 2 for the taxonomical and architectural details in neural networks.

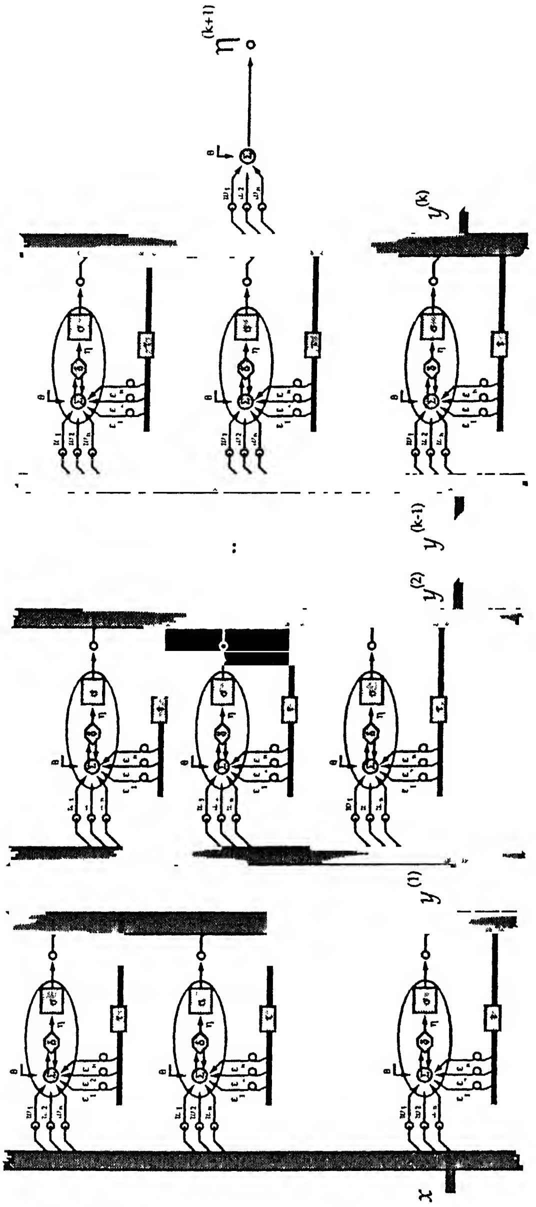


Figure 5.1: Processing hierarchy in type-k neural signal processors.

resentation. Noting that such questions have, indeed, been fruitfully answered in relation to neural networks, I will now rephrase existing statements (applicable to type-1 processors) in the notation adopted in this thesis. Some minor generalizations of the statements have also been attempted along with the rephrasing. Also note that ${}^{k}\mathfrak{N}$ denotes the space of functions realized by type-k neural signal processors. For convenience of analysis, I denote

$${}^{k}\mathfrak{N}(t) = \left\{ \mathfrak{y}^{(k)}(\underline{x}, t) \mid \underline{x} \in \Re^{n} \right\}, \text{for all } k = 1, 2, \ldots\ ; t \in \Re_{0,+},$$

to identify the class of functions realized by a neural signal processor, as a consequence of possible evolution, at time t, ${}^{k}\mathfrak{N}(t) \subseteq {}^{k}\mathfrak{N}$: non-evolutionary processors are characterized by ${}^{k}\mathfrak{N}(t)$ being invariant in t, and for this class of processors, ${}^{k}\mathfrak{N}(t) = {}^{k}\mathfrak{N}$, for all $t \in \Re_{0,+}$. In a similar fashion ${}^{k}\mathfrak{N}_{\underline{m}^{(k)}}(t)$ will be used to denote the class of evolutionary neural functions realized by processors whose nodes are as specified in $\underline{m}^{(k)}$.

Sigmoidal functions, the more popular and traditional choice of activation functions, establish one-one correspondences between $\Re$ and $[\zeta_{-}, \zeta_{+}] \subset \Re$, the latter being the range space of (isolated) neurons, and it is of interest to note the following.

𝔓𝔯𝔬𝔭𝔬𝔰𝔦𝔱𝔦𝔬𝔫 **5.1.1** *For every family of continuous functions* $f \cdot \Re^{n} \rightarrow \Re$*, the family being indexed on* $\Re_{0,+}$ *(responses of members of the family are denoted by* $f(\underline{x}, t)$*,* $\underline{x} \in \Re^{n}$*,* $t \in \Re_{0,+}$*), such that the mapping* f*, at* *every point in $\Re_{0,+}$, is surjective (on $\Re$)*

$$\int\limits_{\Re^n} |\sigma(f(\underline{x}, t))|\, d\mu(\underline{x}) = 0 \quad \equiv \quad \mu = 0, \textit{ for any } t \in \Re_{0,+},$$

where, μ is a finite measure on $\Re^n$ and σ is a sigmoidal function.

𝒫ℛ𝒪𝒪ℱ: This statement reduces to the $L^1(\Re)$ norm of σ noting that the support of σ, for all functions $f(\ ,t)$, at every $t \in \Re_+$, is all of $\Re$. Since the hyperbolic tangent function from which the sigmoidal function is derived is not an integrable function, *ie*, $tanh(\ ) \notin L^1(\Re)$, the desired result follows immediately.

□

Noting that type-1 non-evolutionary neural signal processors defined on $\mathcal{E}^n$, the unit hypercube, with $[\zeta_-, \zeta_+] = [0, 1]$ are dense in the space of continuous functions as established by **Cybenko** (1989), the following characterization of type-k neural signal processors, implied by Proposition 5.1.1, is necessary.

𝒯ℋℰ𝒪ℛℰℳ **5.1.1** *If the activation function σ is* sigmoidal *in the weaker sense of*

$$\sigma \cdot \Re \rightarrow [\zeta_-, \zeta_+] \subset \Re$$

such that $\sigma \in C(\Re)$, ie, *σ is continuous, and*

$$\sigma(x) \rightarrow \begin{cases} \zeta_- & as\ x \rightarrow -\infty, \\ \zeta_+ & as\ x \rightarrow +\infty \end{cases}$$

(for some, appropriately chosen, $\zeta_{-}, \zeta_{+} \in \Re$ such that $\zeta_{-} < \zeta_{+}$), then ${}^{k}\mathfrak{N}(t)$, for all $k = 1, 2, \ldots$, and for all $t \in \Re_{0,+}$, is dense, with respect to any finite measure, in $C(\Re^{n})$, the space of continuous n-variate real functions.

𝒫𝓇𝑜𝑜𝒻: This arguments in this proof have been considerably influenced by the proof provided in *op cit* for the case that reduces to the earlier mentioned characterization of type-1 neural signal processors. Observe that

$$
\begin{aligned}
{}^{k}\mathfrak{N}(t) \;\triangleq\; & \Bigg\{ \mathfrak{y}^{(k)} \colon \Re^{n} \times \{t\} \to \Re \;\Big| \\
& \qquad \mathfrak{y}^{(k)}(\underline{x}, t) = \sum_{j=1}^{m_k} w_j^{(k+1)} \sigma_{j(k)}^{(\lambda)} (\eta_{j(k)}^{(k)} \underline{x}, t) - \theta^{(k+1)} \Bigg\} \\
& \subset C(\Re^{n}),\, \underline{x} \in \Re^{n}, \text{for all } t \in \Re_{+}, \text{ and } k = 1, 2, \ldots,
\end{aligned}
$$

where, $\eta^{(k)}(\underline{x}, t)$ (with appropriate subscripting indices) is given by Equation 5.1 (*p.* 238). ${}^{k}\mathfrak{N}(t)$, $k = 1, 2, \;.\;.\;, t \in \Re_{0,+}$, is a linear subspace of the linear space $C(\Re^{n})$, and incorporates several cases, *ie*, that of deterministic processors in steady-state, processors with additive, or shunting, dynamics with possible stochastic interpretation to the activation functions, *etc*. The claim is that the closure ${}^{k}\overline{\mathfrak{N}}(t)$, of ${}^{k}\mathfrak{N}(t)$, is all of $C(\Re^{n})$, $k = 1, 2, \ldots, t \in \Re_{0,+}$. As the subsequent argument is independent of k, the type number of neural signal processors, and t, the temporal index, quantifications on k and t will not be indicated unless absolutely necessary.

Let, if possible, ${}^{k}\overline{\mathfrak{N}}(t)$, a closed proper subspace of $C(\Re^n)$, be different from $C(\Re^n)$. By the Hahn-Banach theorem (*cf*, **Kreyszig**, 1978; **Rudin**, 1986) for a bounded linear functional, say f, defined on ${}^{k}\overline{\mathfrak{N}}(t)$, and a sub linear functional f_s, on $C(\Re^n)$ satisfying $|f(\mathfrak{y}^{(k)})| \leq |f_s(\mathfrak{y}^{(k)})|$ $\forall\, \mathfrak{y}^{(k)} \in {}^{k}\overline{\mathfrak{N}}(t)$, there exists a bounded linear extension, say $\tilde{f}$, from ${}^{k}\overline{\mathfrak{N}}(t)$ to $C(\Re^n)$ such that $\tilde{f}(\mathfrak{y}^{(k)}) = f(\mathfrak{y}^{(k)})\ \forall\, \mathfrak{y}^{(k)} \in {}^{k}\overline{\mathfrak{N}}$, and $|\tilde{f}(\mathfrak{y}^{(k)})| \leq f_s(\mathfrak{y}^{(k)})$ $\forall\, \mathfrak{y}^{(k)} \in C(\Re^n)$. In particular, if we choose f to be 0 on all of ${}^{k}\mathfrak{N}$ and ${}^{k}\overline{\mathfrak{N}}$, then we can expect an extension $\tilde{f} \neq 0$ on $C(\Re^n)$.

By the Riesz representation theorem (*op cit*) this bounded linear functional, $\tilde{f}$ is expressed as the (functional) inner product

$$\tilde{f}(h) = \int\limits_{\Re^n} h(\underline{x}, t)\, d\mu(\underline{x})\,,\ \forall t \in \Re_{0,+},$$

for some (signed) measure μ of bounded variation on $\Re^n$ and for all $h \in C(\Re^n)$. In view of the nature of the members of ${}^{k}\mathfrak{N}(t)$, and as the coefficients of the linear combination are not all zero, it is essential that

$$\int\limits_{\Re^n} |\sigma(\eta_{j(k)}^{(k)}(\underline{x}, t))|\, d\mu(\underline{x}, t) = 0,\ \forall t \in \Re_{0,+},$$

for all instances of $\eta_{j(k)}^{(k)} \in C(\Re^n)$. However, when sigmoidal activation functions are used, as already established in Proposition 5.1.1 (*p*. 242), this condition implies that $\mu = 0$, *ie*, the functional $\tilde{f} \equiv 0$ on $C(\Re^n)$, a situation contradicting the one we are interested in, thereby establishing the denseness of ${}^{k}\mathfrak{N}(t)$ in $C(\Re^n)$, for all $t \in \Re_{0,+}$.

□

COROLLARY TO THEOREM **5.1.1** *With sigmoidal activation functions, and arbitrary compact subsets $C \subseteq \Re^n$, ${}^k\mathfrak{N}(t)$, $\forall\, k = 1, 2, \ldots$, and for all $t \in \Re_{0,+}$, is dense in $C(C)$ with respect to any finite measure on $C \subseteq \Re^n$.*

The proof of the above statement is on the same lines as that for Theorem 5.1.1 (*p.* 243) except that Riesz representation theorem on compact sets (**Kreyszig**, 1978, *p.* 227) is invoked to establish the contradiction and, hence, is not being detailed. Note that in this discussion, $\Re^n$ refers to an embedding of the (non-null) observation space (which is the input space $\mathcal{X}$ augmented–through a Cartesian product–with the fraction of the output space fed back through lateral interactions and recurrent connections) and $\Re_{0,+}$ allows for a specification of the time index t at which all computation, via the neural processor, is considered.

It is important to note that the above theorem, characterizing the existence of representation in (multi-layered) neural signal processors, requires the activation function σ to have the specific property that

$$\int_{\Re^n} |\sigma(\eta^{(\ell)}(\underline{x}, t))| \, d\mu(\underline{x}) \neq 0 \text{ for all finite measures } \mu \neq 0, \forall t \in \Re_{0,+}, \tag{5.4}$$

ie, the L^1 norm of σ evaluated over all instances of $\eta^{(\ell)}$, for each ℓ, $\ell = 1, 2, \ldots, k$, be non-vanishing, which indicates that the density property is not a feature of the sigmoidal function alone, and that we can expect several other types of activation functions resulting in a similar representation potential. However, as indicated by **Leshno, Ya Lin,** *et al* (1994), denseness of ${}^k\mathfrak{N}(t)$ is assured if and only if the activation functions σ are not algebraic polynomials (almost everywhere). The property indicated in Equation 5.4 is not a tautology noting that a measurable function, say h, allows us to define a measure, say ϕ, in terms of another, say μ:

$$\int_{\mathcal{X}} g(x)\, d\phi(x) = \int_{\mathcal{X}} g(x)\, h(x)\, d\mu(x)\,, \forall g \in C(\mathcal{X})\,, \mathcal{X} \subseteq \Re^n,$$

and in view of this relation (*cf*, **Rudin**, 1986) $\int_{\mathcal{X}} g(x)\, d\phi(x) = 0$ if functions g and ϕ (as decided by h and μ) have mutually disjoint supports. **Cybenko** (1989) terms functions satisfying the property indicated in Equation 5.4 as being *discriminatory*.

Denseness in representation provided by neural signal processors is not restricted to the space of continuous functions nor to sigmoidal activation functions as indicated below.

𝒫ℛ𝒪𝒫𝒪𝒮ℐ𝒯ℐ𝒪𝒩 **5.1.2** *(Refer Hornik's theorems on universal approximation in neural networks reproduced in* ***Leshno, Ya Lin,*** et al, *1994.)*

1. *If the activation function σ is bounded and different from a constant, for any finite measure μ, ${}^k\mathfrak{N}(t)$, for all $k = 1, 2, \ldots$, and for all $t \in \Re_{0,+}$, is dense in $L^p(\mu)$, $1 \le p \le \infty$.*

2. *If the activation function σ is continuous, bounded and non-constant, then for arbitrary compact subsets $C \subseteq \Re^n$ ${}^k\mathfrak{N}(t)$, for all $k = 1, 2, \ldots$, and for all $t \in \Re_{0,+}$, is dense in $C(C)$ with respect to uniform distance.*

The above two statements in Proposition 5.1.2 reaffirm the sense of representation indicated in Theorem 5.1.1 (*p.* 243) with sigmoidal activation functions. Before closing this preliminary discussion on representational capacity, I draw attention to the fact that some of the investigations pertaining to 'three-layer-sufficiency' for the representation of continuous functions relate to statements of type-2 neural signal processors. Of particular interest are works based on a theorem, related to the representation of multivariate functions, due to **Kolmogorov** (1957b) (refined later by **Sprecher**, 1965) and **Arnold** (1957) (see also **Hecht-Nielsen**, 1987c; **Girosi & Poggio**, 1991; **Kůrková**, 1992; **Lagunas, Pérez-Neira**, *et al*, 1993 and **Kovačec & Ribeiro**, 1993).

In these investigations the networks suggested have the form

$$y(\underline{x}) = \sum_{q=0}^{2n} \chi_q(\sum_{p=1}^{n} \pi_{pq}(x_p)) \text{ for all } \underline{x} \in \mathcal{E}^n.$$

While this form, on inspection, is immediately seen to have a similarity with type-2 neural signal processors, differences exist. One difference is in the fact that it is uncommon, in neural signal processors, to subject the inputs to decision-making (*ie*, functions π) without any preprocessing – this has, incidentally, led some researchers to comment that Kolmogorov's theorem is not relevant to neural networks. In § 5.4 (*p.* 284) I will expand on the representational capacity of neural signal processors and, in this effort, will provide an interpretation to Kolmogorov's theorem which will be suited to an appreciation of signal processing with neural networks.

5.2 Functional Nature of Neural Signal Processors

Neural signal processors of type-k in the family ${}^k\mathfrak{N}_{\underline{m}^{(k)}}(t)$ establish functions, for all $t \in \Re_{0,+}$, from $\Re^{m_0}$ to $\mathcal{Y} \subset \Re^{m_{\overline{k+1}}}$, where $m_0 \equiv n$ is the number of incident, or observed, channels. The functions are, in general, indexed by $t \in \Re_{0,+}$, a variable with the connotations of time: under steady state considerations, this index is dropped for notational convenience. In the sequel, the nature of functions established by neural signal processors and their characterization will be in focus. Extension of the ensuing statements to situations wherein neural signal processing includes topological spaces, *ie*, when the input pattern space $\mathcal{X}$ (and, consequently, the observation space) is a manifold embedded in the topological vector space $\Re^{m_0}$ and the output pattern space $\mathcal{Y}$ is a manifold embedded in the topological vector space $\Re^{\overline{m_{k+1}}}$, though possible, has not been included in the scope of this work.

PROPOSITION **5.2.1** *Each processor in* ${}^k\mathfrak{N}_{\underline{m}^{(k)}}(t)$*, for all* k*,* $k = 1, 2, \ldots$ *, and for all* $t \in \Re_{0,+}$ *defines an operator from* $\Re^{m_0}$ *to* $\Re^{m_{\overline{k+1}}} \supseteq \mathcal{Y}$*.*

PROOF: Recall the definition of neural signal processors: a type-1 neural signal processor consists of three operational stages. In the first stage, *ie*, measurements, $\eta^{(1)}_{j^{(1)}}$, $j^{(1)} = 1, 2, \ldots, m_1$, are evaluated from the presented pattern $\underline{x}$ and, if relevant, the past history of processor operation. The second stage enables discrimination on the measurements, through nonlinear evaluation of $\eta^{(1)}_{j^{(1)}}$, to get corresponding values (discriminates) $y^{(1)}_{j^{(1)}}$, $j^{(1)} = 1, 2, . . , m_1$. Finally, the discriminates are linearly combined to provide a response $\mathfrak{y}^{(1)}$, which has the interpretation of concept, category, estimate, *etc*, depending on the context in which neural signal processors are being discussed.

It is quite easy to see that the measurement functions $\eta^{(1)}_{j^{(1)}}$, $j^{(1)} = 1, 2, \ . . , m_1$, are, each, by definition, indexed collection of operators from $\Re^{m_0}$ to $\Re$ (indexing being over $\Re_{0,+}$) by virtue of their being realized through an accumulation (in the discrete sense as summation and in the continuous sense as integration) of inner products and results of operators $(b^{(1)}_{j^{(1)}})$ acting on $\eta^{(1)}_{j^{(1)}}$, modulated by the result of operators $(a^{(1)}_{j^{(1)}})$ acting on $\eta^{(1)}_{j^{(1)}}$. Similarly, the discrimination functions $\sigma^{(1)}_{j^{(1)}}$, $j^{(1)} = 1, 2, \ldots, m_1$, are, each, operators from $\Re$ to $\Re$, and the aggregation of $\mathfrak{y}^{(1)}$ through $y^{(1)}_{j^{(1)}}$, $j^{(1)} = 1, 2, . . \ , m_1$, is a linear operator from $\Re^{m_1}$ to $\Re$. The function of a type-1 neural signal processor, at every time instant in $\Re_{0,+}$, being realized through an appropriate composition of measurement, discrimination and aggregation operators, is, obviously, an operator from $\Re^{m_0}$ to $\Re$. A neural signal processor of type-1, with vector valued response (the output vector $\underline{\mathfrak{y}}$ having m_2) is, on the same lines, an operator from $\Re^{m_0}$ to $\Re^{m_2}$, for all $t \in \Re_{0,+}$.

Given an appropriate k-layered stacking of measurement and discrimination stages, ultimately with a vector valued response having $m_{\overline{k+1}}$ elements, we observe, again, from the definition, that a type-$\overline{k+1}$ neural signal processor is realized by first subjecting the $m_{\overline{k+1}}$ outputs (of the given k-layered ensemble) to the operators $\sigma^{(k+1)}_{j^{(k+1)}}$, $j^{(k+1)} = 1, 2, \ . m_{\overline{k+1}}$, followed by the linear operator aggregating $\mathfrak{y}^{(k+1)}$ from $y^{(k+1)}_{j^{(k+1)}}$, $j^{(k+1)} = 1, 2, \ldots m_{\overline{k+1}}$. This organization of processing immediately suggests that, the function of a type-k neural signal processor mapping from $\Re^{m_0}$ to $\Re^{m_{\overline{k+1}}}$, at every $t \in \Re_{0,+}$, in view of being derived as a composition of operators, is, trivially, an operator.

□

Continuity of the operator is assured only if, in all the participating nodes, the activation function (σ) is continuous. Boundedness of the operator induced by neural signal processors follows, trivially, from boundedness of the activation functions employed.

Proposition **5.2.2** *(Type number additivity.) Every vector-valued neural signal processor of type-k,* $\underline{\mathfrak{y}}^{(k)}(\underline{x}, t) \in {}^{k}\mathfrak{N}_{\underline{m}^{(k)}}(t)$, *for all k,* $k = 2, 3, \ . \ ,$ *has a decomposition in terms of neural signal processors of lower type numbers*

$$\underline{\mathfrak{y}}^{(k)}(\underline{x}, t) = \left(\underline{\mathfrak{y}}^{(k-k_1)} \circ \underline{\mathfrak{y}}^{(k_1)}\right)(\underline{x}, t),$$

for some $k_1 = 1, 2, \ldots \ , k-1$, *for all* $\underline{x} \in \Re^{m_0}$ *and* $t \in \Re_{0,+}$.

Proof: Note that in a neural signal processor of, say type-k_1, $k_1 = 1, 2, \ldots, k-1$, the aggregation of outputs $\underline{\mathfrak{y}}^{(k_1)}$ from discriminates $y^{(k)}_{j^{(k_1)}}$, $j^{(k_1)} = 1, 2, \ldots, m_{k_1}$, is expressible as the linear transformation

$$\underline{\mathfrak{y}}^{(k)}(\underline{x}, t) = \mathbf{W}^{(k+1)} \underline{y}^{(k)}(\underline{x}, t) - \underline{\theta}^{(k+1)},$$

where,

$$W^{(k+1)} = \left[\underline{w}_1^{(k+1)}, \underline{w}_2^{(k+1)}, \;..\, \underline{w}_{m_{k+1}}^{(k+1)}\right]^{\mathsf{T}},$$

and

$$\underline{\theta}^{(k+1)} = \left[\theta_1^{(k+1)}, \theta_2^{(k+1)}, .\; \theta_{m_{k+1}}^{(k+1)}\right]^{\mathsf{T}}$$

Consequently, a processor described as

$$\underline{\mathfrak{y}}^{(k-k_1)}(\underline{\mathfrak{y}}^{(k_1)}(\underline{x}, t), t) \triangleq \left(\underline{\mathfrak{y}}^{(k-k_1)} \circ \underline{\mathfrak{y}}^{(k_1)}\right)(\underline{x}, t)$$

is identical, in functionality, with a processor in ${}^{k}\mathfrak{N}$ noting that the outer processor (of type-$\overline{k-k_1}$) evaluates[8] $\underline{\tilde{s}}^{1}_{\tilde{\jmath}^{(1)}} W^{(k_1+1)} \underline{y}^{(k)}(\underline{x}, t)$, $\tilde{\jmath}^{(1)} = 1, 2, \;..\,, \tilde{m}_0 \equiv m_{\overline{k_1+1}}$ as components of the first layer of measurements These evaluations do not alter the linearity of inner products and allow for a simple concatenation of the constituent layers of measure-discriminate-aggregate stages.

□

This decomposition, a simple consequence of the definition, is, in general, non-unique and allows the following to be anticipated.

PROPOSITION **5.2.3** *Every neural signal processor of type-k, for all k, $k = 1, 2, \ldots$, has a decomposition involving k type-*1 *neural signal processors,* ie,

$$\forall \underline{\mathfrak{y}}^{(k)} \in {}^{k}\mathfrak{N}_{\underline{m}^{(k)}}(t) \;\exists\, \underline{\mathfrak{y}}_1^{(1)}, \underline{\mathfrak{y}}_2^{(1)}, \ldots, \underline{\mathfrak{y}}_k^{(1)} \in {}^{1}\mathfrak{N}_{\underline{m}^{(1)}}(t)$$

$$\textit{such that } \underline{\mathfrak{y}}^{(k)} = \underline{\mathfrak{y}}_1^{(1)} \circ \underline{\mathfrak{y}}_2^{(1)} \circ \cdots \underline{\mathfrak{y}}_k^{(1)}.$$

[8] For reasons of notational clarity, entities relevant to the processor in ${}^{\overline{k-k_1}}\mathfrak{N}(t)$ are shown with the accent ˜.

𝒫ℛ𝒪𝒪ℱ: This statement follows directly from Proposition 5.2.2 by taking $k_1 = 1$ and recursing on the processors of type larger than 1.

□

(Note that dependence of the response of the neural signal processors on $(\underline{x}, t)$ has not been explicitly shown.)

While the above statements are quite obvious, they do highlight that at an architectural level, neural signal processors of all types are composed of layers of in star-out star neuronal fields[9] and provide a simple, though inadequate, justification for the representational potential of type-k neural signal processors, noting that, in view of function composition, the denseness of ${}^{k}\mathfrak{N}(t)$ in $C(\Re^{m_0})$ (also $L^p(\Re^{m_0})\,[\mu]$) follows from the density theorems for ${}^{1}\mathfrak{N}(t)$. It is to be noted that all neural signal processors of type-k, $k = 2, 3, \ \ldots$, can be viewed as a type-$\overline{k - k_1}$ processor operating on the result of a type-k_1 neural signal processor. This suggests that preprocessing, if any, of the presented signals (patterns) can be sought to be represented in a neural basis.

An organizational feature common to neural signal processors, noting the layers of in star-out star neuronal fields, is that the desired processing is achieved through stages of measure-discriminate-aggregate layers. Each layer, essentially a processor in ${}^{1}\mathfrak{N}(t)$, realizes its function through measurement, discrimination and aggregation stages. In the

[9] In Chapter 2 the notions of in star and out star neurons and neuronal fields have been introduced.

common presentations, measurement and aggregation are captured as linear processors and discrimination is incorporated through the non-linear activation functions: the resultant, from the point of view of classification, categorization and recognition, should, inevitably, be a nonlinear operation so that the ensuing processor is non-trivial. These observations motivate the following.

AXIOM **5.2.1** ***Axiom of Organization.***
A neural signal processor is composed of (layers of) three operational stages: measurement, discrimination and aggregation in that order. Preprocessing, if any, (preceding, or incorporated in, the measurement) is sought to be represented in a neural basis. Measurements are effected on an observation space constructed as the Cartesian product of the input space and a relevant subspace of a union of the space of responses of the distinct layers.

Though the above axiom suggests the necessity of three operational stages, these need not be distinct. Processing schemes wherein the non-linear nature of the operation results from measurements rather than discrimination are known in the literature (*eg*, **Davidson & Hummer**, 1993). Recall that nonlinearity in the processor functionality is considered essential in concept realization through decision making. In view of the interpretation that outputs of neural signal processors are concepts the following statement is a consequence of Proposition 5.2.2 (*p*. 251).

PROPOSITION **5.2.4** *A given situation of concept realization (*ie, *function mapping) can be achieved with different choices of 'intermediate concepts,' the latter also being known as 'sub-concepts.'*

PROOF: From Proposition 5.2.2, we note that a concatenation of type-k_1 neural signal processors with type-$\overline{k-k_1}$ processors results in type-k neural signal processors, the resulting structure being equivalent to a type-$\overline{k-k_1}$ processor acting on the response $\underline{\mathfrak{y}}^{(k_1)}(\underline{x},t)$ of a type-k_1 processor. If responses $\underline{\mathfrak{y}}^{(k_1)}(\underline{x},t)\,(\triangleq \boldsymbol{W}^{(k+1)}\underline{y}^{(k)}(\underline{x},t) - \underline{\theta}^{(k+1)})$ are now interpreted as concepts, we note that in view of the evaluation $\underline{\tilde{s}}^{(1)}_{\tilde{j}^{(1)}}\boldsymbol{W}^{(k_1+1)}\underline{y}^{(k)}(\underline{x},t)$, $\tilde{j}^{(1)} = 1,2,\ldots,\tilde{m}_0 \equiv m_{\overline{k_1+1}}$, in the components of the measurements, uniqueness of the final response is decided by the uniqueness of the product vector $\underline{\tilde{s}}^{(1)}_{\tilde{j}^{(1)}}\boldsymbol{W}^{(k_1+1)}$, $\tilde{j}^{(1)} = 1,2,\ \ldots,\tilde{m}_0 \equiv m_{k_1+1}$, rather than that of either $\underline{\tilde{s}}^{(1)}_{\tilde{j}^{(1)}}$ or $\boldsymbol{W}^{(k_1+1)}$.

□

This situation is to be contrasted with the multiplicity of representations shown in Chapter 4: while in § 4.1 non-uniqueness of the solutions for terms like $\underline{\tilde{s}}^{(1)}_{\tilde{j}^{(1)}}\boldsymbol{W}^{(k_1+1)}$, *ie*, product of weights, was in focus, Proposition 5.2.4 refers to the non-uniqueness in realizing the weights through products. A consequence of the above statement, in particular with a signal processing perspective, is that computational cognitive science can, at best, discuss *operationally sufficient* models and not *logically necessary* models. The possibility of allowing given situations of concept representation using different intermediate concepts is reflected, at an organizational level, in terms of the feed-through interconnection strengths $\underline{s}$ being composed by aggregation weights of one layer and measurement weights of the succeeding layer, leading to an alternative definition for neural signal processors (with unit lateral delays τ)

DEFINITION **5.2.1** *Neural signal processors of type-k, ${}^{k}\mathfrak{N}(t)$, are given by the functional form*

$$\begin{aligned}
\dot{\eta}^{(\ell)}_{j^{(\ell)}}(\underline{x},t) &= a^{(\ell)}_{j^{(\ell)}}(\eta^{(\ell)}_{j^{(\ell)}}(\underline{x},t))\Big[b^{(\ell)}_{j^{(\ell)}}(\eta^{(\ell)}_{j^{(\ell)}}(\underline{x},t)) \\
&\qquad + \underline{w}^{(\ell)}_{j^{(\ell)}}\ \underline{\mathfrak{y}}^{(\ell-1)}(\underline{x},t) + \underline{\epsilon}^{(\ell)}_{j^{(\ell)}}\ \underline{\mathfrak{y}}^{(\ell)}(\underline{x},t-1)\Big], \qquad (5.5a) \\
y^{(\ell)}_{j^{(\ell)}}(\underline{x},t) &= \sigma^{(\ell)}_{j^{(\ell)}}(\eta^{(\ell)}_{j^{(\ell)}}(\underline{x},t)), \qquad (5.5b) \\
\mathfrak{y}^{(\ell)}_{i^{(\ell)}}(\underline{x},t) &= \underline{v}^{(\ell)}_{i^{(\ell)}}\ \underline{y}^{(\ell)}(\underline{x},t) - \theta^{(\ell)}_{i^{(\ell)}}, \qquad (5.5c)
\end{aligned}$$

for all $j^{(\ell)} = 1, 2, \;.., m_\ell$, *for some* $m_\ell = 1, 2, .\;.;$

for all $i^{(\ell)} = 1, 2, \;.., \mathrm{m}_\ell$, *for some* $\mathrm{m}_\ell = 1, 2, .\;\;;$

$\ell = 1, 2, \quad, k,$

where, η, y, $\mathfrak{y}$, $\underline{x}$, t, σ, a, b, s, ϵ *and* θ *have the same interpretations as in* Equation 5.1 (*p.* 238) *and* $\mathfrak{y}^{(0)}(\underline{x},t) \equiv \underline{x}$, $\forall t \in \Re_{0,+}$. *Weights* $\underline{w}$ *and* $\underline{\epsilon}$ *are associated with measurement and* $\underline{v}$ *with (concept) aggregation;* $\underline{w}^{(\ell)}_{j^{(\ell)}} \in \Re^{\mathrm{m}_{\ell-1}}$ *and* $\underline{\epsilon}^{(\ell)}_{j^{(\ell)}} \in \Re^{\mathrm{m}_\ell}$, *for all* $j^{(\ell)} = 1, 2, \ldots, m_\ell$ *and* $\underline{v}^{(\ell)}_{i^{(\ell)}} \in \Re^{m_\ell}$, *for all* $i^{(\ell)} = 1, 2, \ldots, \mathrm{m}_\ell$, $\ell = 1, 2, .\;., k$, *where,* m_ℓ *denotes the number of decision units participating in layer* ℓ *and* m_ℓ *denotes the number of (sub) concepts realized in layer* ℓ, $\ell = 1, 2, \;.\;,$ *with* $\mathrm{m}_0 \equiv n$. $\underline{\theta}^{(\ell)}$, $\ell = 1, 2, \;.,$ *denotes the biases necessary in realizing the relevant numerical assignments corresponding to the (sub) concepts,* $\underline{\theta}^{(\ell)} \in \Re^{\mathrm{m}_\ell}$.

Characterization of neural signal processors is best understood in terms of the functions (operators) induced between the manifolds $\mathcal{X}$ and $\mathcal{Y}$. Type-1 (deterministic, non-evolutionary) processors (${}^1\mathfrak{N}$) with hard-limiting activation functions have been shown to (see *eg*, **Lippmann**, 1987) effect a separation of the input space through (hyper) planes. Networks of such processors have also been shown to partition the input space in terms of convex and/or non-convex regions depending on the number of layers in the network. I will now address the issue of processor characterization in a slightly general framework, basically to arrive at a specification of the operator induced (from $\mathcal{X}$ to $\mathcal{Y}$) by a type-k neural signal processor.

Theorem **5.2.1** *Measurements $\eta^{(k)}$ in neural signal processors of type-k,* ie, *members of ${}^k\mathfrak{N}(t)$, $k = 1, 2, \ldots$, partition the input manifold $\mathcal{X}$, of dimensionality n, in terms of manifolds of dimension no more than $n-1$: in situations wherein the activation functions have stretches of constancy, this dimension is no less than $n-2$.*[10]

Proof: $\dot{\eta}^{(\ell)}_{j^{(\ell)}}$, for all $j^{(\ell)} = 1, 2, \ldots m_\ell$, $\ell = 1, 2, \ldots k$, having its continuity inherited from that of the abstract amplification function a, the abstract translation function b and the activation function σ, the continuity of $\eta^{(\ell)}_{j^{(\ell)}}$ depends on a and b corresponding to the processing node $j^{(\ell)}$ in layer ℓ and a, b and σ of all processing nodes in layers 1 to $\overline{\ell-1}$: in

[10] This statement suggests the interesting possibility of using, as candidates of activation, functions whose preimages, in the domain, are of fractional dimension In this thesis, however, this issue has not been taken up

the popular neural models, *ie*, models of additive as well as shunting dynamics, continuity of a and b is assured in all nodes and the activation functions are allowed to have up to countable discontinuities. Thus the measurements, η, will be considered as continuous functions over $\Re^n$, these functions are indexed over $\Re_{0,+}$ (through the variable t)

Noting that $\eta^{(\ell)}_{j(\ell)}$, for the admissible values of $j^{(\ell)}$ and ℓ, refers to time-indexed spatial measurements, it is simple to see that these are, indeed, manifolds[11] (*ie*, surfaces) over the input space. Continuity of η in the temporal index variable (t) reflects the nature of evolution of the measurements over the input space. It is now simple to see that the measurements η at all layers, nodes and points $t \in \Re_+$ induce manifolds in the input space and the collection, in the sense of a set union, of such manifolds over the possible assignments to $\eta^{(\ell)}_{j(\ell)}(\cdot, t)$ describes the entirety of the input space for each admissible value of ℓ, $j^{(\ell)}$ and t: the individual manifolds for any given values of ℓ, $j^{(\ell)}$ and t are themselves disjoint. The claim regarding the dimension of the manifolds induced in the input space by measurements η will be established using mathematical induction on layering (type number).

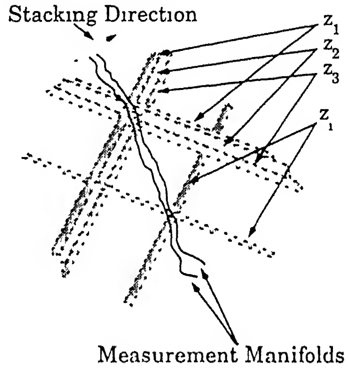


Figure 5.2: Nature of η

[11] A manifold is a geometrical structure that is homeomorphic to $\Re^n$.

Verification

When we consider isolated neurons, *ie*, trivial neural signal processors of type-k for all $k = 1, 2, \ldots$, the claim is immediately apparent as the measurements η induce a partitioning of an n-dimensional input space in terms of (hyper) planes, each of dimension $\overline{n-1}$ and of zero measure with respect[12] to the Lesbegue measure of n dimensions. The activation function σ alters only the manner of appreciating the manifolds induced on the input space by measurements and not the manifolds *per se*: the manifolds induced on the input space as a result of activation functions, on members in the manifolds of measurement, will be termed manifolds induced by decisions. In the case of activation functions that are continuous the dimension (and measure) of the manifold is not altered. Activation functions that have stretches of constancy (*eg*, hard-limiting functions) regroup the manifolds induced by measurement to ensure that the resulting manifolds being continuous unions[13] of $\overline{n-1}$-dimensional mutually disjoint manifolds are of finite (and non-zero) measure and a dimension no less than $\overline{n-2}$. It is now simple to see that the same arguments hold to the measurements of type-1 neural signal processors as such processors are, operationally,

[12] In the rest of this proof the measure will always be taken to be Lesbegue measure on n-dimensions This aspect will not be indicated explicitly.

[13] Note that while the manifolds induced by measurements need only $\overline{n-1}$ distinct basis vectors for a complete description, the manifolds induced by decisions, when the activation functions have regions of constancy, need all of n distinct basis vectors, however, the entire scope of only $\overline{n-1}$ of these basis vectors is used for description: variation along the remaining basis vector is restricted to a (compact) proper subset of the total extent of variation.

no different from being an array of neurons, each being identical in functionality to an isolated neuron, the outputs of which are linearly combined in an attempt to realize the desired processor

Inference

Consider the measurements $\eta^{(\ell)}$. These measurements are evaluated as linear combinations of the concepts in a type-$\overline{\ell-1}$ neural signal processor and thereby the decisions of a neural network of $\overline{\ell-1}$ layers. Noting that the measurement of a point $\underline{x}$ in the input manifold $\mathcal{X}$ is defined only when the decisions of every node in the $\overline{\ell-1}$ layer network participating in the synthesis of $\eta^{(\eta)}$, *ie*, the manifold induced in $\mathcal{X}$ by the distinct assignments to $\eta^{(\ell)}$ is an intersection of $m_{\overline{ell-1}}$ decision regions–the number of processing nodes in layer ℓ, for all ℓ, has been denoted earlier by m_ℓ–and this region, if it exists, has a dimension given by $max(0, n - m_{\overline{\ell-1}})$, wherein the manifolds induced by the decisions taken in a network of $\overline{\ell-1}$ layers has been assumed to be $\overline{n-1}$. The manifold induced in $\mathcal{X}$ by assignments to $\eta^{(\ell)}$ being continuous unions of such regions, the implication of these manifolds being of dimension no larger than $\overline{n-1}$ is immediately seen. By an identical reasoning it can be established that the dimension of the manifolds induced in $\mathcal{X}$ by assignments to $\eta^{(\ell)}$ is constrained, on the upper side, to be one less than the smallest dimension of the manifolds induced in $\mathcal{X}$ by the participating decisions.

In the event the activation function is continuous, it is immediately apparent that the manifolds induced by $\eta^{(\ell)}$ are homeomorphic to (hyper) planes of identical dimension: this aspect has been indicated in the illustration. However, when the activation function has regions of constancy–for simplicity I assume that such regions are observed in the activation functions of all processing nodes–the intersections of the manifolds of decisions in the network of $\overline{\ell - 1}$ layers will be convex regions, thereby, the manifolds induced in $\mathcal{X}$ will be a chaining of such (local) convex regions, the chaining being along a manifold which is no different from that induced by measurements when continuous activation functions are involved. The dimension of this union of convex regions is easily seen to be no less than $\overline{n - 2}$ when all manifolds induced by decisions of the $\overline{\ell - 1}$ layered network, on which the realization of $\eta^{(\ell)}$ is based, are of a dimension no less than $\overline{n - 2}$.

Conclusion

A verification of the claim for type-1 neural signal processors and the assurance of the validity of this result for a type-ℓ processor conditional on its validity for type-$\overline{\ell - 1}$ processors suffices to establish the stated claim. Note that this proof has not needed an explicit use of the node indices $j^{(\ell)}$ and the temporal index t.

□

Note that in the above theorem the partitioning of the input manifold by measurement functions is of a varying (adaptive) nature when recurrence or lateral interaction is involved. The past history of (intermediate) concepts when incorporated in a measurement of ıncident concepts (*ie*, inputs) serves to revise the translation effected by the function b and, thereby, while the dimensionality of the members of the input space partitioning is unaffected, the manner in which partitioning evolves in neural signal processors that incorporate recurrence and/or lateral interaction. Evolution in the partitioning on the input manifold has been the basis (in schemes suggested in the literature) of incorporating search in neural networks. In order to appreciate the nature of manifolds induced by the measurement functions of neural signal processors, the following is introduced (*cf*, **Lawson**, 1974; **Itô**, 1987).

DEFINITION **5.2.2** *A foliation*[14] *of codimension q (alternatively dimension $p = n - q$) on an n-dimensional manıfold M, $0 \leq q \leq n$, ıs a family $\mathfrak{F} = \{L_\alpha \mid \alpha \in \mathcal{A}_{\mathfrak{F}}\}$ of arc wise connected subsets, called* leaves, *of M with the following properties:*

(i) $L_\alpha \cap L_{\alpha^\circ} = \emptyset$ *if* $\alpha \neq \alpha^\circ$, $\alpha, \alpha^\circ \in \mathcal{A}_{\mathfrak{F}}$.

(ii) $\bigcup\limits_{\alpha \in \mathcal{A}_{\mathfrak{F}}} L_\alpha = M$.

[14] More precisely, this is termed a codimension q class C^r-foliation of M The notion of foliations, essentially complex geometrical structures, has also been considered by **Lawson** (1974) and **Itô** (1987) on structures other than manifolds

> A manifold is, roughly speaking, a space locally modeled on affine space, and a sub manifold is a subset locally modeled on an affine subspace In this spirıt, a foliated manifold is a manifold modeled locally on an affine space decomposed into parallel affine subspaces (**Lawson**, 1974).

(iii) Every point in M has a neighborhood $\mathcal{U}$ and a system of local, class C^r coordinates $\underline{x} = (x_1, x_2, \ldots x_n) : \mathcal{U} \mapsto \Re^n$ such that for each leaf $\mathcal{L}_\alpha$, $\alpha \in \mathcal{A}_\mathfrak{F}$, the components of $\mathcal{U} \cap \mathcal{L}_\alpha$ are described by the equations $x_{p+1} =$ constant, $..x_n =$ constant.

Every leaf of $\mathfrak{F}$ is an $(n-q)$ dimensional sub manifold of M. A simple example of a foliation is the collection of (hyper) planes defined in an isolated neuron by an operation described in Equation 3.1a. Figure 5 3 (p. 264) illustrates the notion of foliations. In the definition of foliation suggested by **Lawson** (1974) as well as **Itô** (1987), characteristics of the set $\mathcal{A}_\mathfrak{F}$ which supports the indexing of leaves has not been mentioned. As explained in the following, in the context of neural networks, this set is of paramount importance and I will refer to the set $\mathcal{A}_\mathfrak{F}$ as the *stem*[15] of the foliation $\mathfrak{F}$. For convenience, the set $\mathcal{A}_\mathfrak{F}$ will be considered to be of dimension q, the codimension of the foliation $\mathfrak{F}$. The nature of partitions induced in the input space by the measurement functions η motivates the following.

Axiom 5.2.2 *Axiom of Measurement.*

A neural signal processor, through the measurement functions in each of the processing (decision making) nodes, induces a foliation, of codimension at least one, in the input manifold. This foliation forms the basis of synthesizing (approximating) the desired level curves of the function.

[15] A more appropriate terminology for the set $\mathcal{A}_\mathfrak{F}$ is *stalk* of a foliation However, the term 'stalk' is used in the theory of sheaves (**Tennison**, 1975) to mean a subspace of the manifold and not an index set for leaves

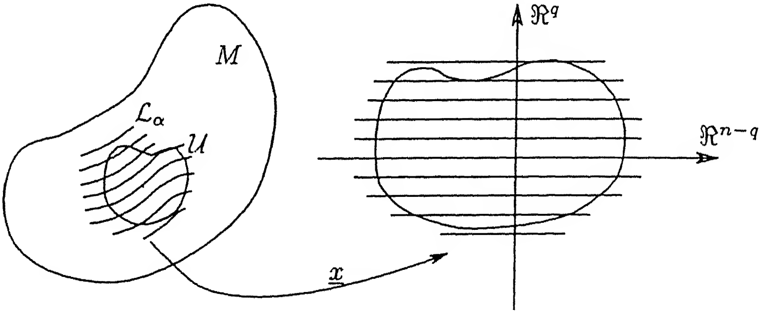


Figure 5.3: Illustration of a foliation on a manifold

Though, in this thesis, the foliation induced by isolated neurons is not considered to be more complicated than that of a partition through (hyper) planes, such a general geometric structure has been incorporated into the axiom of measurement for the following two reasons.

1. The polynomial neurons of **Cover** (1965), higher order neurons of **Spirkovska & Reid** (1992), morphological neurons of **Davidson & Hummer** (1993) and neurons with functional links (**Pao**, 1989) partition the input manifold in terms of (hyper) surfaces rather than (hyper) planes. Figure 5.4 illustrates the foliation induced in these non-conventional neurons: the foliation is considered on $\Re^2$, a two dimensional space in which the input space is embedded.[16]

[16] In Figure 5.4 the foliation induced by polynomial neurons are indicated in (a) and (b), morphological neurons in (c) and (d), a higher order neuron in (e), a neuron with functional links in (f) and a neuron which imposes elliptical basis functions (a generalization of radial basis functions) in (g)

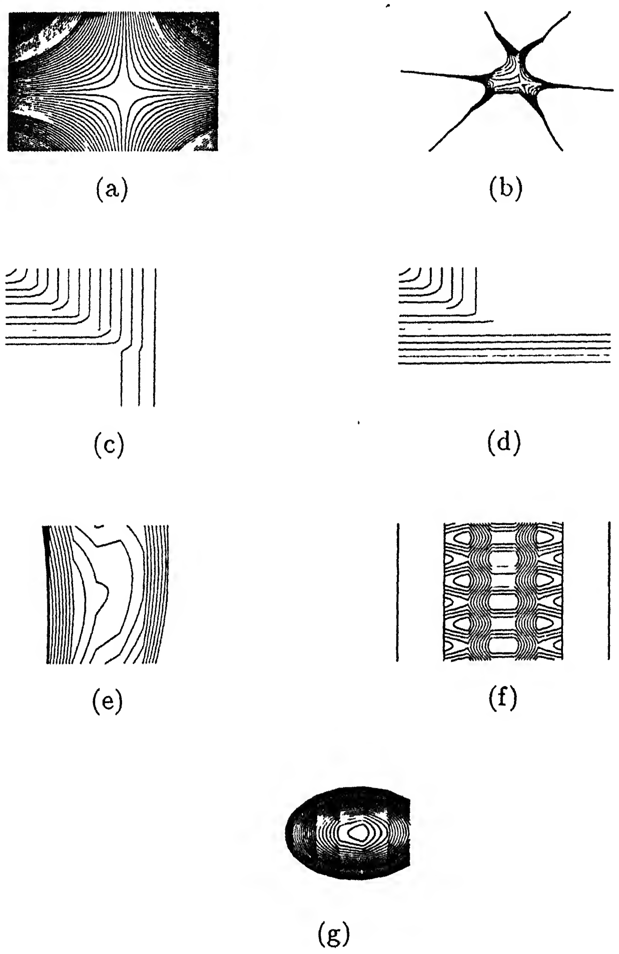


Figure 5.4: Foliation on $\Re^2$ in non-conventional neurons: (a) $\eta(\underline{x}) = 2x_1 + 3x_2 - 2x_1x_2$, (b) $\eta(\underline{x}) = (2x_1 + 3x_2 + 2)(3x_1 - 8x_2 + 12)(7x_1 - x_2 + 1)$, (c) $\eta(\underline{x}) = (2 \wedge x_1) \vee (5 \wedge x_2)$, (d) $\eta(\underline{x}) = (5 \wedge x_1) \vee (-2 \wedge x_2)$, (e) $\eta(\underline{x}) = 2x_1^2 + 5x_2^3 + 3x_1x_2 + x_1 + 3x_2$, (f) $\eta(\underline{x}) = sin(x_1) + tan^2(x_2)$, (g) $\eta(\underline{x}) = (x_1 + 2)^2 + (x_2 - 5)^2$

2. Though the the leaves of the foliation induced on an input manifold by isolated neurons of the type indicated in Equation 3.1 (*p*. 110) are (hyper) planes, the nonlinear nature of activation functions (especially when sigmoidal functions are in use) imply that the leaves of foliations induced by processing nodes in layered neural networks are (hyper) surfaces, the 'curvature' increasing, in general, with the depth of layering

In a foliation on an input manifold, the leaves can have multiple components, each component is, in general, a connected region: Figure 5.4 (a), (b) and (f) indicate leaves which have multiple components. The characterization of multi-layered neural networks (with hard-limiting activation function) provided by **Lippmann** (1987) is really a characterization of the leaves in the foliation on the space of input patterns induced by measurement functions.

Leaves of the foliation in isolated neurons are convex regions with a single component (*ie*, (hyper) planes). The leaves in type-1 neural signal processors are piece-wise (hyper) planar with a single component that is, in general, not convex (however, the union of an uncountably large number of such leaves can still be convex) and the leaves in neural signal processors that are of a type larger than unity have multiple components, each component being piece-wise (hyper) planar.

Piece-wise (hyper) planarity in the leaves introduces a 'curvature,' thereby, allowing the leaves of type-k neural signal processors, for all values of k, $k = 1, 2, \ldots$, to be closed: the closed nature of leaves is related to the representation of local features. (Figure 5.4 (f) and (g) indicate leaves of foliation that have components that are closed.) Superiority of approximation with non-conventional neurons, as claimed in the literature, is easily traced to the presence of multiple components and 'curvature' in the leaves of the foliation.[17]

Measurements, in neural signal processors, provide the discriminants which, through operations of the same genre as comparison, form the basis of decisions presented, to the external world, as neural response. The discriminants are specified by the stem of the foliation induced on the input manifold by the measurement functions. Discriminatory functions which provide the mechanism of associating measurements on the incident inputs to neural action or categories, regardless of the specific details (see § 2.2 for different types of activation functions), serve to establish equivalences between distinct measurement values, thereby, associating distinct members of an input space with common clusters. the nature of the clusters is decided by the number of components and the nature of closedness in the leaves of the foli-

[17] Isolated neurons of the type indicated in Equation 3 1 (equated with processors capable of realizing linear separable dichotomies), as shown in Chapter 3, represent a fraction of total number of functions possible on discrete input spaces, this fraction vanishes as the dimensionality of the input space increases. Contrasting this with the situation in an interconnected schema of Turing Machines wherein each processor is allowed to realize all possible functions on the discrete space, an enquiry on the nature of basic processing units necessary and/or sufficient in an ensemble to accommodate a realization, with considerations of efficiency, of the desired (information) processing task is prompted. This enquiry, however, has not been included in the scope of this thesis

ation induced by measurement functions and the kinds of association between leaves through the activation functions.

The essential purpose of discrimination is to effect a foliation of the input manifold on the basis of the foliation induced by measurements: the leaves of the foliation due to discrimination are unions of leaves of the foliation due to measurement. Equivalently, discrimination effects a transformation between the stems of foliations without actually altering the system of local coordinates in the neighborhoods of points in the input manifold. Let ${}^m\mathfrak{F} = \{{}^mL_\alpha \mid \alpha \in \mathcal{A}_{{}^m\mathfrak{F}}\}$ denote the foliation on the input manifold due to measurements and ${}^d\mathfrak{F} = \{{}^dL_\alpha \mid \alpha \in \mathcal{A}_{{}^d\mathfrak{F}}\}$ denote the foliation on the input manifold due to discrimination.

Noting that the activation function establishes a transformation of $\mathcal{A}_{{}^m\mathfrak{F}}$, the space of discriminants, into $\mathcal{A}_{{}^d\mathfrak{F}}$, the space of decision labels, *ie,* σ: $\mathcal{A}_{{}^m\mathfrak{F}} \longrightarrow \mathcal{A}_{{}^d\mathfrak{F}}$, the leaves of the foliation due to discrimination are given, in terms of the leaves of the foliation due to measurement functions, as

$$\forall \alpha \in \mathcal{A}_{{}^d\mathfrak{F}} \; {}^dL_\alpha = \bigcup_{\alpha' \in \sigma^{-1}(\alpha)} {}^mL_{\alpha'},$$

where, $\sigma^{-1}(\alpha)$ refers to the preimage of $\alpha \in \mathcal{A}_{{}^d\mathfrak{F}}$ in $\mathcal{A}_{{}^m\mathfrak{F}}$ under the activation function σ. In this light the following characterization of the requirement of discriminatory functions, in addition to the specification made earlier in connection with representation potential, is of interest.

Axiom **5.2.3** ***Axiom of Discrimination.***

A neural signal processor, through its discriminatory functions, renews the foliations, induced on the input space by the measurement functions, through a transformation, of the stems of the foliations, with at least one of the following properties:

1. *alter the indexing of leaves to retain distinctness in a finite non-zero number of local regions of the input space,*
2. *introduce multiple components in the leaves,*
3. *associate, to at least one component of a leaf of the foliation due to discrimination, uncountably many leaves of the foliation due to measurement.*

Re-foliations provide the basis for establishing equivalences between members (elements) of the input space in ways not possible through the chosen measurement functions.

As a reordering of foliations, *ie* a recreation of the partition on the input space, is the key objective of discrimination, it is essential that the functions that accomplish this task be different from linear: typical choices for discriminatory functions incorporate reordering through comparison with one or more thresholding parameters. The role of discrimination is one of deciding on features provided by measurements and the functional nature of discrimination functions is to involve a

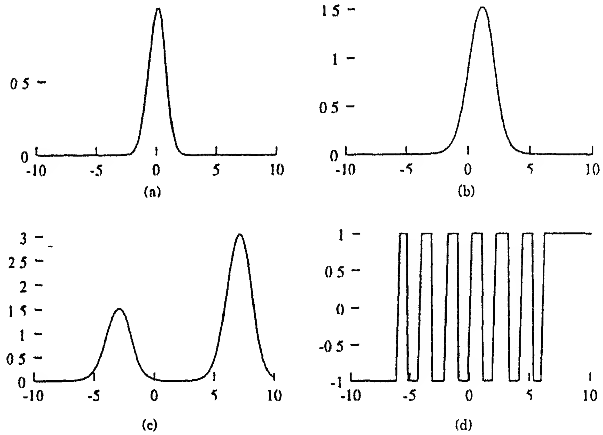


Figure 5.5: Illustration of admissible activation functions: (a) $\sigma(\xi) = e^{-\xi^2}$, (b) $\sigma(\xi) = \sigma_b(\xi) \triangleq tanh(\xi) - tanh(\xi - 2)$, (c) $\sigma(\xi) = \sigma_b(\xi + 4) + 2\sigma_b(\xi - 6)$, (d) $\sigma(\xi) = sgn(\xi) \forall \xi \notin [-6, +6]$, and $\sigma(\xi) = sgn(sin(\xi))$, otherwise, $\xi \in \Re$

comparison of features discovered in the input patterns with templates of the features being tested for: this restricts the choice of activation functions, in the context of measurements inducing foliations of codimension one, to piece-wise[18] monotonic functions, each monotonic segment providing a graded comparison. Figure 5.5 illustrates a few examples of the admissible activation functions: in this illustration, the stem of the foliation due to measurements is identified with $\Re$.

[18] The notion of piece-wise monotonic functions, though, vacuous in the sense that every function is expressible as an alternation of monotonically increasing and monotonically decreasing segments, is being used to explicitly indicate the alternating stretches of monotonic variation

The comparison is considered as representing operations of a sentential calculus of (an appropriate) logic incorporating uncertainty when the discriminatory function is continuous. Multiplicity in the number of monotonic segments introduces unconnectedness (due to multiplicity of components) in the leaves of the foliation induced, in the input space, by decisions and finiteness of the number of components assures that the re-ordering is not trivial. Discontinuities in the discriminatory (*ie*, activation) functions, favored when crisp categorization is needed, are to be finite in order that the categorization problem be computable (see **Hopcroft & Ullman**, 1989 for the notion of computability). Representation of crisp categories necessitate non-strict monotonic variation in the discriminatory functions (indicated by property 3 of the axiom of discrimination).

Arguments in the literature related to the superiority of approximation and function realization provided by measurement functions that induce a foliation whose leaves are not linear subspaces of the input manifold and activation functions that incorporate locality, *eg*, radial basis functions ($y(\underline{x}) = e^{-\|\underline{x}-\underline{x}_t\|^2}$, where $\underline{x}_t$ is the template of the pattern being tested for), provide a characterization of the foliation induced by measurement and not that by discrimination noting that despite non-monotonicity in the activation function, the one-sidedness of the discriminants restricts the leaves of the foliation due to discrimination to have identical number of components as in the leaves in the foliation due to measurement. It can be conjectured, at this stage, that a choice of activation functions that have more than one region of locality, thereby inducing the leaves of the foliation due to discrimination to have multiple components, reduces the number of layers and the number of processing nodes required for a given realization problem of smooth functions.

It is important to appreciate that concepts are aggregates of features (reflected by a discrimination of measurements on the input space), the aggregation process being dependent on the nature of the concept· *ie*, taxonomical or complexive. The objective of aggregation is the inverse of measurement in the sense that while measurement derives features from input instances or examples, aggregation is to synthesize responses from decisions. Inputs are of the same genre as their responses in the same way that pattern features are of the same genre as decisions taken on these features and, hence, the following aspect of aggregation, together with the preceding axioms, would provide a characterization of neural signal processors.

AXIOM **5.2.4** ***Axiom of Aggregation.***
A neural signal processor, through its aggregation function, synthesizes (or approximates) the level regions of processor response through a foliation on the Cartesian product of the stems of foliations on the input space due to discrimination. Concepts, in neural signal processors, are identified with the level regions of processor response.

Thus if, in a certain level of discrimination, the foliations induced by the processors are denoted by ${}^{d}\mathfrak{F}_i \quad \{{}^{d}_{i}L_{\alpha_i} \mid \alpha_i \in \mathcal{A}_{{}^{d}\mathfrak{F}_i}\}$, $i \quad 1, 2, .\ .m$, then the leaves of the foliation ${}^{a}\mathfrak{F} = \{{}^{a}L_{\alpha'} \mid \alpha' \in \mathcal{A}_{{}^{a}\mathfrak{F}}\}$ on the space of input patterns due to aggregation are given by

$$\forall\alpha' \in \mathcal{A}_{{}^{a}\mathfrak{F}} \; {}^{a}L_{\alpha'} = \bigcap_{\underline{\alpha} \in {}^{s}L_{\alpha'}} {}^{d}_{i}L_{\alpha_i}, \tag{5.6}$$

where ${}^{s}L_{\alpha'}$ denotes a leaf in the foliation, ${}^{s}\mathfrak{F}$, on $\mathcal{A}_{{}^{d}\mathfrak{F}_1} \times \mathcal{A}_{{}^{d}\mathfrak{F}_2} \cdots \mathcal{A}_{{}^{d}\mathfrak{F}_m}$ and $\underline{\alpha} = [\alpha_1, \alpha_2, .\ .\alpha_m]^{\top}$. Note that in the foliation ${}^{a}\mathfrak{F}$ on the collection of input patterns, described through the foliation ${}^{s}\mathfrak{F}$ on the Cartesian product of the stems of the foliations on the input space due to discrimination, $\mathcal{A}_{{}^{a}\mathfrak{F}} (\equiv \mathcal{A}_{{}^{s}\mathfrak{F}})$ refers to the collection of responses of the neural signal processor From the arguments leading to Proposition 5.2.4 (*p.* 255) it can be seen clearly that the foliation induced on the input space due to measurement functions operating on the responses of a type-k neural signal processor, for any value of k, $k = 1, 2, \ .\ .$, has a structure similar to that indicated in Equation 5.6.

Figure 5.6 illustrates a foliation induced on the input space due to an aggregation of foliations due a discrimination of a partitioning (foliation) provided by (hyper) planes: the activation function has been assumed to be sigmoidal and the foliations due to discrimination are considered from two distinct processors. As indicated in this figure, properties 1 and 3 of the activation function given by the axiom of discrimination together with the foliation on the Cartesian product of the stems of the foliations due to discrimination introduce a 'curvature'

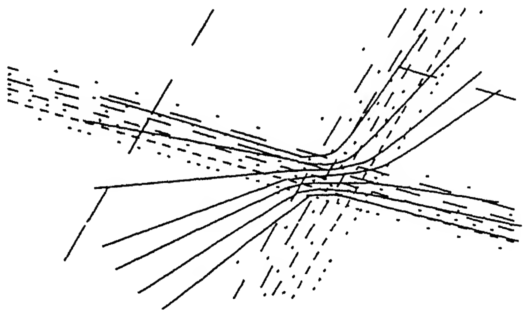


Figure 5.6: Illustration of a foliation due to aggregation

in the leaves of the foliation on the input space due to aggregation even if the leaves of the foliation on the input space due to measurement do not exhibit any 'curvature.'

The preceding discussion, on the one hand, suggests the adequacy, for representation, of a processing scheme involving a layered organization of weighted linear combinations of the responses of sigmoidally transformed weighted linear combinations of incident patterns (or intermediate concepts), while, on the other hand, triggers the possibility of novel processing schemes. Note that the statement of adequacy has been, more formally, stated in Theorem 5.1.1 (*p*. 243). A suggestion of processing schemes alternate to that indicated in Equation 5.1 (*p*. 238), however, is not included in the scope of this thesis.

Before closing this brief presentation of the functional nature of neural signal processors, I draw attention to the fact that the axioms of neural signal processing are not restricted to only those stated in the foregoing. However, it is important to realize that these axioms characterize the information processing aspect of neural signal processor. The context in which such processing is handled needs to be specified by separate axioms that specify the choices to be made in the input pattern space (vector space, topological manifold, symbolic space, *etc*), decision space (discrete or continuous) and space of concepts (numerical or symbolic).

Such axioms have, intentionally, been left open to accommodate generality in the conceptual framework of neural signal processors, though, in this thesis, I have been using the common choices (with minor generalizations) of an input space embedded in the n-dimensional space $\Re^n$, where n denotes the number of input channels, decisions restricted to a (compact) subspace of $\Re^m$, where m denotes the number of decisions being sought on patterns from the input space $\mathcal{X}$ and a space of concepts (*ie*, responses of the neural signal processor) embedded in $\Re^{\mathrm{m}}$, where m is the number of elements in the processor response $\underline{\mathfrak{y}}$. All the spaces are considered as (metric/normed) vector spaces of appropriate dimensions: this choice stems from the present considerations of relevance in discussions of neural signal processors.[19]

[19] A more complete picture of the perspective and computational power (in comparison with other decision making frameworks like Turing Machines) of the paradigm of neural signal processors will be obtained when all the spaces are considered as varieties of category theory (**Hartshorne** (1977)), a topic outside the scope of this thesis

5.3 Operational Interpretation of Neural Signal Processors

Neurons, and neural signal processors, till now characterized as operators between finite dimensional spaces will, in the ensuing discussion, be extended to slightly general domains, specifically function spaces: consequently, extension of the neural processing paradigm to functional and operator spaces, and thereby, to the space of neural signal processors, can be expected.[20] This exercise is aimed at seeking a means for unifying existing neural processing architectures in terms of operations familiar in the context of signal processing. For simplicity of presentation, only the formal neuron model corresponding to steady state solution under additive dynamics will be considered, noting that extensions to other categories of neural models follow similar reasoning.

I begin with the functional structure of an isolated (hypothetical) neuron. The formal model of a neuron, as discussed in § 2.2, describing the steady state solution (when the input is unvarying in time) is given, with the usual conditions, by

$$y(\underline{x}) = \sigma(\sum_{i=1}^{n} w_i x_i - \theta) \equiv \sigma(\underline{w}\cdot\underline{x} - \theta)\,. \tag{5.7}$$

It is common knowledge that inner product operation is available over Euclidean spaces as well as function spaces, and this feature of the inner

[20] In this thesis, however, the scope of the neural processing paradigm has been restricted to pattern spaces, with the associated interpretation of (in general, discrete) signal spaces Extensions to functional, operator (specifically, neural processor spaces ${}^{k}\mathfrak{N}$, $k = 1, 2, \ldots$), functor spaces and other varieties of category theory (**Krishnan**, 1981) will be reported elsewhere

product operation will govern the definition of neurons over function spaces Thus, the manifolds $\mathcal{X}$ and $\mathcal{Y}$ will, in the present article, be considered as function spaces and for reasons of distinction be denoted by $\mathfrak{X}$ and $\mathfrak{Y}$ respectively. Similarly, weights are drawn from a function space denoted by $\mathfrak{W}$ as distinct from $\mathcal{W}$ used in the case of neurons over Euclidean spaces.

Definition 5.3.1 *The formal model*[21] *of an isolated neuron on a function space $\mathfrak{X}$ weighted by a function in $\mathfrak{W}$ is given by*[22]

$$\eta(x) = \langle w(\gamma), x(\gamma) \rangle - \theta, \qquad (5\ 8a)$$

$$y(x) = \sigma(\eta(x)), \qquad (5.8b)$$

where, $w \in \mathfrak{W}$ and $x \in \mathfrak{X}$ are functions defined on the (entire) real number field, $\gamma \in \Re$ is the continuous valued index of the functions x and w, $\theta \in \Re$ is the threshold, and $\langle \cdot, \rangle$ is the inner product operation between functions:

$$\langle w(\gamma), x(\gamma) \rangle = \int_{\Re} d\gamma \, w(\gamma) \, x(\gamma) \, . \qquad (5.9)$$

(Note that as this discussion is based only on real valued functions x and w, complex conjugation in the inner product between functions has not been incorporated.)

[21] In order to extend the formal model of neurons with dynamics to function spaces, we need to replace inner products between vectors by inner products between functions.

[22] Note that in neurons, and in neural signal processors, the measurement, discrimination, and aggregation (not incorporated in neurons) stages are still operators, though between function spaces.

It is worthwhile to observe that the above definition subsumes the formal model of neurons on finite dimensional spaces noting the elementary principle that functional inner product subsumes vector inner product. We note that the extension of neural definition to function spaces does not, however, improve the separation capability in comparison with neurons defined on Euclidean spaces. Neural signal processors over function spaces are defined similar to those over Euclidean (pattern) spaces (see § 5.1 and § 5.2) with the specific difference that vector inner products are replaced by inner products between functions, and thereby inherit all the taxonomical, architectural, and representational peculiarities discussed earlier. The response of a type-k neural signal processor will, for notational convenience, be continued to be denoted by $\mathfrak{y}^{(k)}$. I introduce the following in order to facilitate an appreciation of the operational character of neural signal processors

DEFINITION 5.3.2 *Neural signal processors of type-k, $k = 1, 2, \,.\,,$ are defined by the (informal) operator equations*

$$\begin{aligned}
\dot{\eta}^{(\ell)}_{\xi^{(\ell)}}(x,t) &= a^{(\ell)}_{\xi^{(\ell)}}(\eta^{(\ell)}_{\xi^{(\ell)}}(x,t)) \Bigg[b^{(\ell)}_{\xi^{(\ell)}}(\eta^{(\ell)}_{\xi^{(\ell)}}(x,t)) \\
&\quad + \int_{\Gamma_{\ell-1}} d\mu^{(\ell-1)}_{w}(\gamma^{(\ell-1)})\, \boldsymbol{K}^{(\ell)}_{w}(\xi^{(\ell)}, \gamma^{(\ell-1)})\, \mathfrak{y}^{(\ell-1)}_{\gamma^{(\ell-1)}}(x,t) \\
&\quad + \int_{\Gamma_{\ell}} d\mu^{(\ell)}_{\epsilon}(\gamma^{(\ell)})\, \boldsymbol{K}^{(\ell)}_{\epsilon}(\xi^{(\ell)}, \gamma^{(\ell)})\, \mathfrak{y}^{(\ell)}_{\gamma^{(\ell)}}(x,t-1) \Bigg], \qquad (5.10a)\\
y^{(\ell)}_{\xi^{(\ell)}}(x,t) &= \sigma^{(\ell)}_{\xi^{(\ell)}}(\eta^{(\ell)}_{\xi^{(\ell)}}(x,t)), \qquad (5.10b)
\end{aligned}$$

$$\mathfrak{y}^{(\ell)}_{\gamma^{(\ell)}}(x,t) \;=\; \int_{\Xi_\ell} d\mu_v^{(\ell)}(\xi^{(\ell)})\, K_v^{(\ell)}(\gamma^{(\ell)},\xi^{(\ell)})\, y^{(\ell)}_{\xi^{(\ell)}}(x,t) - \theta^{(\ell)}_{\gamma^{(\ell)}}, \qquad (5.10c)$$

where, $\xi^{(\ell)} \in \Xi^{(\ell)}$ is the index for the collection of decisions in layer ℓ, $\gamma^{(\ell)} \in \Gamma^{(\ell)}$ is the index for the collection of (sub) concepts in layer ℓ, $\ell = 1, 2, \ .., k$, η, t, σ, a, and b have the same interpretation as in Equation 5.1, $x \in \mathfrak{X}$ is the (real valued) input function, θ is the (real valued) threshold function, y is an operator whose values are decision functions, $\mathfrak{y}$ is operator whose values are the neural signal processor responses, ie,

$$\mathfrak{y}^{(\ell)} \cdot \Gamma^{(\ell)} \longrightarrow {}^k\mathfrak{N}(t),\ \forall \ell,\ \ell = 1, 2, \ldots, k,$$

and $\mathfrak{y}^{(0)}(x,t) \equiv x$, $\forall t \in \Re_{0,+}$. K_w denotes the 'feed-through measurement kernel,'[23] $K_w^{(\ell)}(\xi^{(\ell)},\cdot) \in \mathfrak{W}$, K_ϵ denotes the 'measurement kernel due to lateral interaction,' $K_\epsilon^{(\ell)}(\xi^{(\ell)},\cdot) \in \mathfrak{E}$ and K_v denotes the 'aggregation kernel,' $K_v^{(\ell)}(\xi^{(\ell)},\cdot) \in \mathfrak{V}$, where $\mathfrak{W}$, $\mathfrak{E}$, and $\mathfrak{V}$, are collections of admissible weighting functions. $\forall \xi^{(\ell)} \in \Xi^{(\ell)}$ $\forall \gamma^{(\ell)} \in \Gamma^{(\ell)}$ $K_w^{(\ell)}(\xi^{(\ell)},\gamma^{(\ell)}) = w^{(\ell)}_{\xi^{(\ell)}}(\gamma^{(\ell)})$, $K_\epsilon^{(\ell)}(\xi^{(\ell)},\gamma^{(\ell)}) = \epsilon^{(\ell)}_{\xi^{(\ell)}}(\gamma^{(\ell)})$, and $K_v^{(\ell)}(\xi^{(\ell)},\gamma^{(\ell)}) = v^{(\ell)}_{\xi^{(\ell)}}(\gamma^{(\ell)})$, where, w, ϵ, and v are weighting functions (or vectors) defined on lines similar to that of Definition 5.2.1. Concept definition spaces $\Gamma^{(\ell)}$, $\ell = 0, 1, 2, \ldots, k$, and decision definition spaces $\Xi^{(\ell)}$, $\ell = 1, 2, \ldots, k$, are, each, allowed to be either continuous or discrete (and finite). In case any of these spaces is discrete, integration over the corresponding spaces is to understood in the sense of summation. μ_w, μ_ϵ, and μ_v, with appropriate layering indices, denote the measures with respect to which integration

[23] The feed-through measurement kernel is equivalent to the matrix $\boldsymbol{W}$ considered in the proof of Proposition 5.2.2 (p. 251)

is carried out. The propagation delays τ between processing nodes are assumed, for convenience, to be unity while considering lateral interaction and zero for feed-through connections.

Note that closure of neural signal processing over the space of concepts is implicit in the definition due to the assignment $\mathfrak{y}^{(0)}(x,t) \equiv x$, $\forall t \in \Re_{0,+}$. The discussion presented in the foregoing, though initiated in the context of neurons defined on function spaces can be easily translated to the case of processors realized on Euclidean spaces noting the conceptual similarities in the two cases: however, for reasons of convenience, the translation to neurons on Euclidean spaces is not being presented. In the case of processors realized on Euclidean spaces ($\Re^n$), the integral transforms are of the discrete variety.[24]

A few remarks related to the aggregation kernels are in order. If we appreciate the relationship between decisions and concepts and consider the possibility of deriving the (logically) necessary and/or sufficient decisions given concepts, then the process, essentially one of de-aggregation, is the inverse of measurement and neural signal processors, in this sense, associate the integral transforms of measure-

[24] Though, in all of the preceding discussion, the inputs have always been considered as linear arrays, this being generalized to functions, the operational aspect of neural signal processing is not restricted to such an interpretation By an appropriate choice of variables, the kernels of the integral transforms can be revised to have interpretations of processing multi-dimensional signal (images) without the need for an interpretation in terms of arrays of functions Such an interpretation would be essential in situations wherein the features and concepts have to interpreted as related to sub-images of the incident image. Such interpretations have been suggested in the literature.

ment and de-aggregation. However, as inverses of the integral transforms are not assured, and are not being imposed either, such a view, though accurate, would not be operationally feasible. For this reason, the paradigm of neural signal processing is being considered as non-linear associations between the integral transforms of measurement and aggregation. Neural signal processors between function spaces are characterized as in the following.

Theorem **5.3.1**

1. *Measurement and aggregation operations in neural signal processors are integral transforms.*
2. *In type-*1 *processors, $^1\mathfrak{N}(t)$, for all $t \in \Re_{0,+}$, these transforms are linear.*
3. *Measurement operations of type-k neural signal processors, viewed with respect to the input, are non-linear integral transforms: non-linearity in the kernel is related to 'curvature' in the leaves of the foliation induced on the input function space.*
4. *Activation functions impose point wise non-linear relationships between the integral transforms of measurement and aggregation.*

Corollary to Theorem **5.3.1** *Neural signal processors between finite dimensional spaces are discrete (linear) integral transforms.*

Though these statements are trivial consequences of the above definition, it is important to note that the operational paradigm of representation in neural information processing is as specified in the following.

Theorem **5.3.2** *Information processing in neural signal processors is effected by point-wise nonlinear associations between (nonlinear) integral transforms*

$$\begin{tikzcd}
\eta(t) \colon \Xi_\ell \to \Re \arrow[r, mapsto, "\sigma"] & y(t) \colon \Xi_\ell \to \mathcal{Y} \arrow[d, "K_v^{(k,\ell)}(t)"] \\
x(t) \colon \Gamma_0 \to \mathcal{X} \arrow[u, "{K_w^{(\ell,0)}(t), K_\epsilon^{(\ell)}}"] \arrow[r, Mapsto] & \mathfrak{y}(t) \colon \Gamma_k \to \Re
\end{tikzcd}$$

Neural information processing paradigm is captured effectively by the above display with the understanding that $x(t)$, $\eta(t)$, $y(t)$, and $\mathfrak{y}(t)$ are spatially defined operators indexed in $t \in \Re_{0,+}$ (commonly interpreted as time), and for some *a priori* chosen $\ell \in \{1, 2, .\ \ k\}$, $K_w^{(\ell,0)}(t)$ is the effective feed-through measurement kernel operating between the inputs and layer ℓ, and $K_v^{(k,\ell)}(t)$ is the effective aggregation kernel operating between layer ℓ and the outputs. This immediately allows an appreciation of neural signal processing in the light of conventional signal processing (see § 2.1).

Neural signal processing relates to conventional signal processing in the following ways:

1. While both approaches are operationally rooted in the realization of processors as associations between integral transforms, the important difference is that while conventional signal processing is based on a choice of integral transforms independent of the family of processors being realized and the synthesis of processors is guided by the choices on the mechanism of association between the integral transforms, neural signal processing considers the mechanism of association to be independent of the processor family, the realization being accomplished through a search for the appropriate kernels of the integral transforms, the latter aspect being appreciated as learning. This operational interpretation is strengthened by Kolmogorov's theorem on function representation (see § 5.4).

2. Signal processing viewed as the distinct stages of feature extraction, decision making and signal reconstruction in the conventional approach, it is imperative that the equivalent of aggregation integral transforms be the inverse of the equivalent of measurement integral transform, *ie*, measurements are the same as de-aggregations: this requirement is related to the signals in the input and output space being of similar nature and interpretative content. Neural signal processing, on the other hand, is not restricted to this interpretation and, hence, it is not common to find the kernels of aggregation and measurement integral transforms being compared. It would not be incorrect to suggest that while

the 'basis' functions through which the desired processing functionality is realized is chosen to be invariant in conventional signal processing, these 'basis' functions are synthesized in accordance with the processing requirement, thereby instilling confidence in claims that the neural approach to signal processor realization is well suited as generic representational framework.

3. In conventional signal processing, in particular, approaches that enjoy linearity in the constituent operations, the mechanism of association between integral transforms providing the key to processor realization, it is not uncommon to find this mechanism being identified as the transfer function between signals in the input and output space: the transfer function, in view of being an association between integral transforms, is invariably given an interpretation in the spectral domain. The transfer function approach is not sustainable in the neural signal processing context as the mechanism of association is, in general, invariant to the family of processors being realized and this mechanism is required to exhibit a nonlinear dependence, of outputs on inputs, in order to facilitate categorization.

5.4 Representation in Neural Signal Processors

Neural signal processors, independent of the type number, have been shown, in the foregoing, to approximate continuous functions and operators on compact (measurable) spaces with any desired degree of accuracy. This statement, applicable for functions and operators realized through static as well as dynamic processors, is true, however, only when the *discriminatory* requirement Equation 5.4 (*p.* 246) is met by the activation function σ at all the participating processing nodes. Activation functions that are smooth at the asymptotes and can be characterized as having alternating stretches of monotonic variation easily satisfy this discriminatory property: typical examples are the sigmoidal and Gaussian functions (see § 2.2).

The theorems (see § 5.1) relating to the representation (in the sense of approximation) of continuous functions assure arbitrary degree of approximation accuracy only in the asymptotic situation, *ie*, approximation error decreases as the number of processors increases towards ∞, and remain silent on the character of representation when a finite number of processors are involved, and on the more important issue of specifying the number of processors required in a given processing context. Originally addressed by **Lippmann** (1987), **Hecht-Nielsen** (1987c) and **Baum & Haussler** (1989) in different contexts, with a follow-up by several others investigators, the problem of specifying the size of a network given a specific processing situation does not yet have a common consensus in the solution criterion, and the literature is proliferated with several 'rules of the thumb', or heuristic approaches.[25]

[25] It is worth pointing out that the number of nodes in the 'hidden layer' as a function of the 'dimensionality' of input space (more accurately, number of elements in the input patterns) are debated, quite occasionally, in the Internet discussion forum supported by

Recalling, once again, the theorems of function representation in neural signal processors, the property of denseness in approximation is not restricted to a specific range of processor types, *ie*, for all k, $k = 1, 2, \ldots, {}^{k}\mathfrak{N}(t)$, $\forall t \in \Re_{0,+}$, is dense in the space of continuous functions. This means that from the point of view of function approximation without any restriction on the number of processing nodes, layering is insignificant in the neural processing paradigm. While this observation does not cause much consternation in feed-forward, non-evolutionary networks, an immediate implication in processors supporting evolution in the response (state) through recurrence (*eg*, Hopfield circuit) or lateral interaction (*eg*, Kohonen layer), and enjoying the luxury of arbitrarily large number of discrimination nodes, is that the relevant attractor (or fixed point) should be reached in one computational step. Such a claim is not sustainable unless the class of processors under consideration is trivial, and, hence, it is imperative that a closer look be given to the idea of layering in neural networks.

In the following, the issue of representation with finite number of processing nodes in layered neural networks will be in focus. The (finite) processing structures suggested by an application of a function representation theorem due to **Kolmogorov** (1957b) are cursorily looked into. A proof of this function representation theorem due to **Arnold**

the newsgroup `Comp.ai neural-nets` Dominant heuristics suggest that the number of (hidden) nodes in processors with a single decision layer (essentially members of ${}^{1}\mathfrak{N}$) be related to a mean value based on the number of inputs and output elements. both arithmetic and geometric means have been proposed.

(1958) justifies functions realized by neural signal processors to be characterized in terms of level regions, *ie* leaves of the foliation, on the input space. Noting that neural signal processors establish point-wise nonlinear associations between (discrete) integral transforms, the kernels of these transforms are identified with the individual processing stages in the representation scheme central to this theorem. The discussion will be rounded up with remarks on the realization of the kernels.

Kolmogorov (1957b) in a study of Hilbert's 13th problem (*cf*, **Lorentz**, 1962) has shown that continuous real valued multi-variate real functions (on the unit hypercube $\mathcal{E}^n$) have a representation of a form equivalent to

$$f(\underset{\sim}{x}) = \sum_{q=0}^{2n} \chi_q (\sum_{p=1}^{n} \pi_{pq}(x_p)) \,, \text{ for all } \underset{\sim}{x} \in \mathcal{E}^n, \text{ for all } n = 2, 3, . \,. , \qquad (5.11)$$

where, the functions χ and π are continuous real valued real functions defined on $\mathcal{E}^1 \equiv [0, 1]$, and the choice of functions $\{\pi_{pq}\}_{pq}$, (each being monotone increasing, and $\pi_{p_1 q_1} \neq \pi_{p_2 q_2}$, if $(p_1, q_1) \neq (p_2, q_2)$, $p_1, p_2 = 1, 2, .. \; n$, $q_1, q_2 = 0, 1, \ldots 2n$) is *independent* of the class of n-variate functions $\{f\}$ to be represented. The problem of representation is formulated as one involving appropriate choice of functions χ given a specification of the particular function f to be represented.[26]

[26] Observe that the problem of function representation considered by **Kolmogorov** (1957a) is very similar, in an operational sense, to that initiated by **Rosenblatt** (1958) in the study of perceptrons. The perceptrons of Rosenblatt are two-layered neural networks with weight adaptivity allowed only in the second layer The weights of the first layer are chosen to be appropriate for a 'problem domain' and are independent of the specific processing function to be realized A similar correspondence has been established, in the

The similarity in the forms of computational specification in the representation suggested by Kolmogorov, and layered[27] neural networks (*ie*, type-2 neural signal processors), has motivated **Hecht-Nielsen** (1987c), **Kůrková** (1992) and **Kovačec & Ribeiro** (1993), among others, to claim that multi-layered neural networks are capable of representing all continuous functions of interest. However, noting that the processing structure suggested by Kolmogorov recommends decision functions π_{pq} to be applied directly on the inputs x_p, for all stages q (with appropriate quantification on p, and q), *ie*, the incident input patterns are subjected to decision-making without any preprocessing, Kolmogorov's theorem is not readily applicable to characterize the input-output map provided by neural signal processors: this feature has prompted some investigators to comment that Kolmogorov's theorem is not relevant to neural networks. I suggest an interpretation to the function representation theorem by Kolmogorov, in the context of neural signal processing, through the following theorem.

𝔗𝔥𝔢𝔬𝔯𝔢𝔪 **5.4.1** *Scalar processors in* ${}^{k}\mathfrak{N}(t)$, $k = 1, 2, . .$, *for all* $t \in \Re_{0,+}$, *with sigmoidal (or monotone increasing) activation function, having two or more nodes in the first layer of processing (*ie, $m_1 = 2, 3, . . .$*), and the feature vector* $\underline{\eta}(\underline{x}, t)$ *restricted to a bounded (linear) subspace*

literature, between the function representation theorem of Kolmogorov and the CMAC architecture of **Albus** (1975).

[27] In the literature, such networks are also termed as being three-layered when the input terminations are counted as a layer. The prevailing lack of consensus in numbering neural networks, and the compulsions for the present nomenclature have been indicated in Chapter 2.

of $\Re^{m_1}$, isomorphic with $\mathcal{E}^{m_1}$, for all $\underline{x} \in \Re^n$ and $t \in \Re_{0,+}$, represent continuous functions of the form

$$\begin{aligned}\mathfrak{h}^{(k)}(\underline{x}, t) &\triangleq \tilde{\mathfrak{h}}^{(k)}(\eta_1^{(1)}(\underline{x}, t), \eta_2^{(1)}(\underline{x}, t), \ldots \eta_{m_1}^{(1)}(\underline{x}, t)) \\ &= \sum_{q=0}^{2m_1} \chi_q (\sum_{p=1}^{m_1} \pi_{pq}(\eta_p^{(1)}(\underline{x}, t))),\end{aligned}$$

where, functions π satisfy the requirements stated in connection with Kolmogorov's theorem, and χ depend on $\tilde{\mathfrak{h}}$.

This interpretation assures that multi-layered neural networks with a finite number of processors, the number depending on the dimension of the initial (*ie*, first layer) feature space, are capable of providing the desired function representation: in this light, the relevance of Kolmogorov's representation theorem in neural signal processing is in the sense of a statement of representational complexity of concepts, given the features, rather than a suggestion for achieving the representation. However, unlike in conventional approaches to neural signal processing, the desired processing task is to be explicitly decomposed into distinct stages of feature extraction and discrimination.

Concepts $\mathfrak{h}$, described on the input patterns $\underline{x}$, are equivalently appreciated as mappings $\tilde{\mathfrak{h}}$ on the initial feature vector $\underline{\eta}^{(1)}(\underline{x})$, and this interpretation, though useful in the hybrid approach to automated intelligence is restricted to a discrimination on features extracted, from the incident signal, using *a priori* specified hypotheses. The above proposition suggests that, akin to the notion of preservance in processors defined on discrete input spaces (see Chapters 3 and 4), the representation of the input space in the measurements ($\eta^{(1)}$), *ie*, input space foliation due to measurement functions, plays a crucial role in the representational nature of neural signal processors. While the above interpretation assures finiteness in the number of processing nodes, this cannot be used to decide the number of layers required in a neural signal processor.

Similarities, of form, in the function representation schemes of layered neural networks and that in the interpretation offered by Theorem 5.4.1 (*p*. 288) are tempting enough to associate processing stages χ and π with the components of neural signal processors. The function representation claims of **Hecht-Nielsen** (1987c), **Kůrková** (1992) and **Kovačec & Ribeiro** (1993), are, in fact, based on such a comparison of operational forms. While these claims have stressed on identifying χ and π with the activation functions, typically monotonic (as required by Kolmogorov's proof), such identification can only be heuristic rather than a rigorous justification.

Noting that neural signal processors establish point-wise nonlinear associations between integral transforms, it is interesting to identify functions χ and π in terms of the kernels of these transforms and activation functions of the processing nodes. This association is being sought from the point of view of understanding the nature of abstraction involved in the mapping represented by $\mathfrak{y}$ rather than correlate, node by node, the two representation schemes. For simplicity of reasoning, I will consider only feed-forward neural signal processors on finite-dimensional input spaces

PROPOSITION 5.4.1 *The χ and π functions of* Theorem 5.4.1 (p 288) *are related to the aggregation kernels $K_v^{(k,1)}$ and $K_v^{(1)}$, respectively.*

PROOF: Noting that the responses of type-k neural signal processors can be written as

$$\mathfrak{y}_i^{(k)}(\underline{x}) = \sum_{j^{(k)}=1}^{m_k} K_v^{(k,1)}(i, j^{(k)}, \sigma_{j^{(k)}}^{(k)}(\sum_{j^{(1)}=1}^{m_1} K_v^{(1)}(j^{(k)}, j^{(1)})\, \sigma_{j^{(1)}}^{(1)}(\eta_{j^{(1)}}^{(1)}(\underline{x}))))$$

it is not difficult, on a term by term comparison with the representational form in Theorem 5.4.1 (p. 288), to see that the π functions are linearly dependent on the activation functions σ and that the χ functions are nonlinearly dependent on σ, the nature of nonlinearity being decided by $K_v^{(k,1)}$, the kernel effective in the transformation of concepts realized by a single layer neural signal processor into measurements at the final layer of a type-k neural signal processor.

□

In order that the nature of the kernels effected in the transformations across layers is understood, the following is introduced. (See **Hazewinkel**, 1988.)

𝔇𝔢𝔣𝔦𝔫𝔦𝔱𝔦𝔬𝔫 **5.4.1** *A transformation T given by*

$$T(x) = \int_a^b K(t, s, x(s))\, ds$$

where $K(t, s, u)$, $a \leq t, s \leq b$, $-\infty \leq t \leq +\infty$, *is a function such that*

$$g(t) = \int_a^b K(t, s, x(s))\, ds$$

is continuous on $[a, b]$, *for any* $x(s)$ *in* $C([a, b])$, *and ıs nonlinear in* u *is termed a nonlinear Urysohn operator mapping* $C([a, b])$ *into itself.*

A discrete version of this operator can be immediately visualized. From the above definition and the preceding proposition the following is immediately evident.

𝔓𝔯𝔬𝔭𝔬𝔰𝔦𝔱𝔦𝔬𝔫 **5.4.2** *The kernels of measurement* $K_w^{(\ell,0)}$, $K_\epsilon^{(\ell,0)}$ *and the kernel of aggregation* $K^{(k,\ell)}$ *in type-k neural signal processors,* ie, ${}^k\mathfrak{N}(t)$, *belong to the class of kernels of Urysohn operators.*

This statement implies that as a representational paradigm, association between integral transforms, synthesized as the influence of multi-layered neural signal processors, is not vacuous. In addition the Urysohn-Brouwer Lemma (**Hazewinkel**, 1988), an assertion on the possibility of extending a continuous function from a subspace of a topological space to the whole space, in the context of an interpretation, in Chapters 3 and 4, of generalization as function extension, provides the basis for an assurance that learning and generalization, the central issues of neural signal processing, can be easily incorporated in discussions involving processors defined on abstract spaces: this thesis, however, being of limited scope, learning and generalization in processors with abstract neurons will not be taken up.

In § 5.2 I have suggested the plausible axioms for a discourse on representation with neural signal processors. These axioms provide a characterization of the admissible structure in the components of a neural signal processor, *viz*, the activation function and the kernels of the integral transforms of measurement and aggregation. As stated earlier, the motivation for seeking the axioms of neural signal processing – fully recognizing the empirical nature of investigations in neural networks – is to provide a framework that would aid a unified approach in the understanding of the nature of representation in neural networks and related 'automata': the unification, however, has not been included in the scope of this thesis.

Recall the axiom of discrimination and consider the activation functions, other than the sigmoid function (including hard-limiter), that are admissible. As suggested in the illustration in Figure 5.5 (*p*. 270), the admissible activation functions have one or more local 'bumps.' Such activation functions have been suggested, in the literature, to be superior to the sigmoidal function for purposes of approximation. (See **Poggio & Girosi**, 1990, for claims regarding the superiority of *reg-*

ularization networks, also called *radial basis function networks*, over approximation through type-1 neural signal processors with sigmoidal activation function in all the processing nodes.)

In view of Theorem 5.1.1 (*p*. 243) and Proposition 5.1.2 (*p*. 247), all activation functions admissible under the axiom of discrimination share the property that ${}^{k}\mathfrak{N}$, the collection of functions realized in a type-k, $k = 1, 2, \ldots$, neural signal processor that incorporates these activation functions, is dense in the space of continuous functions. **Geva & Sitte** (1992) have suggested a constructive procedure for realizing local functions through a weighted superposition of (domain) translated sigmoid functions: the weight values are mutually negative. A similar scheme has been suggested by **Zhang & Benveniste** (1992) and **Pati & Krishnaprasad** (1993), however, in these schemes the linear combination of (domain) translates of the sigmoid function, together with a (domain) scaling and/or rotation, has been identified with wavelet transforms (**Chui**, 1992; **Daubechies**, 1992). These investigations motivate the following elementary statements.[28]

𝔓𝔯𝔬𝔭𝔬𝔰𝔦𝔱𝔦𝔬𝔫 **5.4.3** *To every neuron defined as in* Equation 5.8 (*p*. 277), *where the activation function σ is continuous and satisfies the axiom of discrimination and the other symbols have the same connotations as*

[28] Though the following statements are being made in the general context of neurons defined on function spaces, equivalent statements corresponding to neurons defined on Euclidean spaces is immediately evident.

*in Definition 5.3.1, there exists a corresponding functionally equivalent type-*1 *feed-forward neural signal processor.*

PROOF: This statement is, simply, a consequence of Theorem 5.1.1.

Recall the processing model in Equation 5.8 (*p.* 277).

$$\begin{aligned} \eta(x) &= \langle w(\gamma), x(\gamma) \rangle - \theta, \\ y(x) &= \sigma(\eta(x)) \end{aligned}$$

The approximability[29] statement of Theorem 5.1.1 suggests that there exist real numbers α_i, w_i and θ_i, $i = 1, 2, \ldots m$, for some appropriate finite (positive integer) value of m, such that

$$\sigma(\xi) = \sum_{i=1}^{m} \alpha_i \sigma_s (w_i \xi - \theta_i) \ \forall \xi \in \Re,$$

where σ_s denotes the sigmoidal activation function. This implies that the response of the neuron is given by

$$y(x) = \mathfrak{y}(x) = \sum_{i=1}^{m} \alpha_i \sigma_s (\eta_i(x)) \,.$$

[29] The notion of 'approximability' is to be understood in the sense of computability (**Hopcroft & Ullman**, 1989) extended to the context of function realization Note that function realization is one of the valid instances of computation Denseness of the space of realized functions in the space of desired functions, as suggested by Theorem 5.1.1 (*p.* 243) and Proposition 5.1 2 (*p.* 247), is really an assurance of the possibility (though, in an asymptotic sense) of realizing any function, in the collection of desired functions, with arbitrary accuracy. Approximability is not the same as computability, however. The difference between these two notions arises due to the fact that no restriction of finiteness of the number of computational steps, *ie*, component functions, is assured by approximability, whereas computability necessitates finiteness in the number of computational steps Further, no component function participating in the approximation is assured to be computable.

Since that the above expression is the operational model of a type-1 neural signal processor with a discrete aggregation kernel, the statement is established noting that the equality between y and $\mathfrak{y}$ is in the L^2 sense and the measurements $\eta_i(x)$ represent the following evaluation:

$$\eta_i(x) = w_i\langle w, x\rangle - w_i\theta - \theta_i,\ i = 1, 2, \ldots m$$

□

Corollary to Proposition 5.4.3 *To every type-k, $k = 1, 2, \ldots$, neural signal processor wherein the activation functions are continuous and satisfy the axiom of discrimination, and the kernels of the integral transforms of measurement and aggregation are linear, there exists a corresponding functionally equivalent neural signal processor, of type no more than k, wherein all activation functions are sigmoidal, the kernels of the integral transforms of measurement and aggregation are linear and the complexity of evolution, if any, is unaltered.*

The above corollary to Proposition 5.4.3 follows from the proposition of type number additivity (Proposition 5.2.2 (*p.* 251)) and Proposition 5.4.3. Note that in the above Proposition and its corollary, the measurement kernel of the type-1 neural signal processor, corresponding to every processing node, is made of weighting functions that are linearly dependent on each other. This situation, wherein the weighting functions (weight vectors) of distinct nodes (of the equivalent neural signal processor) are all in the same 'direction' but have different norms, is complementary to that investigated in Chapter 4 in connection with neural signal processors that are realized with preservance weights in the first layer.

In Chapter 4, the weights of distinct nodes, in the first layer, are restricted to have different 'directions' but the same norm so that a choice of preservance weights corresponding to the (preservance) input space in all the processing nodes is still capable of maintaining distinctness of weights in the distinct nodes. Contrasting this, a representation of the activation function in the space of sigmoidal functions has necessitated the weights of distinct nodes to be in the same 'direction' but have different norms.

Consider the case when a neural signal processor of the kind suggested in Proposition 5.4.3 (*p*. 294) is operative on a (discrete) preservance input space of the kind studied in Chapter 3 with $\underline{w}$, assumed non-null, is the associated preservance weight. This processor is functionally equivalent to an isolated neuron whose activation function is continuous, different from a sigmoidal function, and satisfies the axiom of discrimination. Note that $\underline{w}$ provides the common 'direction' for the weights of the processing nodes in the type-1 neural signal processor.

In Proposition 3.2.13 (*p*. 152) I have shown that for every preservance weight $\underline{w}$ of a discrete (preservance) input space there exists at least one other weight (that is not the negative of $\underline{w}$) which is equivalent to $\underline{w}$, in the sense of a representation of functions on the (preservance) input space. An immediate implication of this equivalence and the nature of representation in neural signal processors is that the neural signal processor suggested by Proposition 5.4.3 suggests a representation scheme that involves superpositions of functions on discrete spaces that are permutations (including scaling) of the preservance input space Put differently, a superposition of functions on discrete spaces that are permutations (with scaling) of a preservance input space is equivalent to a (type-1) neural signal processor operating on the preservance input space with non-sigmoidal, continuous activation functions that satisfy the axiom of discrimination.

Other than the axiom of discrimination, the axioms of measurement and aggregation state, indirectly, the structural requirements on the kernels of the integral transforms of measurement and aggregation, *ie*, the weights associated with the distinct nodes in the neural signal processor. The influence of the axiom of discrimination (and preservance) on the choice of weights (and, thereby, the kernels) was motivated by a realization of the activation function in a neural signal processor. If instead, the measurement functions, η, are realized in a neural signal processor, the following is easily established: the proof of these statements is similar to that for Proposition 5.4.3. In the following statements, though the kernels of measurement integral transforms are assumed to be non-linear (the measurement functions are assumed to be polynomial discriminants), the kernels of the integral transform of aggregation will be assumed to be linear. The processors are all assumed to be non-evolutionary.

PROPOSITION 5.4.4 *To every homogeneous polynomial discriminant, expressed compactly as a product of linear discriminants,*

$$P_j(\underline{x}) = \prod_{i=1}^{j} (\underline{w}_i \cdot \underline{x} - \theta_i),$$

*for every value of j, $j = 1, 2, \ldots,$, $\underline{w}_i \in \Re^n$, $\theta_i \in \Re$, $i = 1, 2, \ldots j$, there exists a corresponding functionally equivalent type-*1 *neural signal processor with sigmoidal activation functions and linear discriminants.*

COROLLARY TO PROPOSITION 5.4.4 *To every high-order neuron defined as*

$$\begin{aligned} \eta(x) &= \sum_{j=1}^{N} \prod_{i_j=1}^{j} \left(\langle w_{i_j}, x \rangle - \theta_{i_j} \right), \\ y(x) &= \sigma(\eta(x)), \end{aligned}$$

*for every value of N, $N = 1, 2, \ldots$, $w_{i_j} \in \mathfrak{W}$, $\theta_{i_j} \imath \Re$, $i_j = 1, 2, \ldots j$, $j = 1, 2, \ldots N$, there exists a corresponding functionally equivalent type-*1 *neural signal processor with sigmoidal activation functions and linear kernels of the integral transforms of measurement and aggregation.*

An incorporation of the representations of the measurement functions η and the activation functions σ, in terms of type-1 neural signal processors involving linear combinations of sigmoidal functions operating on linear discriminants, immediately suggests a processing scheme similar to that described by Equation 5.11 (*p*. 287), *ie*

$$\mathfrak{y}(x) = \sum_{j=1}^{N_1} \beta_j \sigma_s(\sum_{i_j=1}^{N_2} \alpha_{i_j} \sigma_s(\langle w_{i_j}, x\rangle - \theta_{i_j}) - \theta_j) \ \forall x \in \mathfrak{X}, \tag{5.13}$$

the equality is in the L^2 sense. In this scheme, the coefficients α and β and thresholds θ, with appropriate suffixes, are real numbers and the weighting functions w are members of the weight function space $\mathfrak{W}$. N_1 and N_2 are finite positive integers which suggest the number of components that participate in the representation (approximation).

Recursing into Theorem 5.4.1 (*p*. 288) with the attention restricted to the variant of Equation 5.13 for neural signal processors defined on Euclidean spaces, *ie*

$$\mathfrak{y}(\underline{x}) = \sum_{j=1}^{N_1} \beta_j \sigma_s(\sum_{i_j=1}^{N_2} \alpha_{i_j} \sigma_s(\underline{w}_{i_j}\ \underline{x} - \theta_{i_j}) - \theta_j) \ \forall \underline{x} \in \mathcal{X}, \tag{5.14}$$

it is easy to observe the following. Representation in neural signal processors, regardless of the type of activation functions and kernels used for the integral transforms of measurement and aggregation, is in the sense of a realization of the desired function as a linear combination of 'basis' functions, these 'basis' functions themselves being synthesized as sigmoidal transformations of the response of a type-1 neural signal processor. (The synthesis of 'basis' functions has already been related to the realization of the class of Urysohn operators.)

Note that the inner representation in Equation 5.13 and Equation 5.14 refers to a realization of the measurement functions η due to nonlinear measurement kernels and the outer decomposition incorporates the synthesis of activation functions that satisfy the axiom of discrimination through superpositions of sigmoidal transformations of weighted measurements. The weights $\underline{w}_{ij}$ (functions w_{ij}), coefficients α_{ij} and thresholds θ_{ij} reflect the representation of the measurement functions and influence the π functions of the network structure suggested by the function representation theorem due to **Kolmogorov** (1957a). In a similar way, the coefficients β_j and thresholds θ_j incorporate aggregations of neural decisions on measurements (on the incident patterns or signals), the neural synthesis of the decision mechanisms (activation functions) and influence the χ functions of thc network structure in Theorem 5.4.1.

Borrowing the interpretations of the χ and π functions in the representation scheme of Theorem 5.4.1, it is evident that the role of every neural signal processor is to realize the given (signal) processing functionality by synthesizing 'basis' functions that, in turn, reflect the synthesis of the required mechanisms of measurement and decision-making: these 'basis' functions operate on 'features' derived from the incident pattern (signal). The architecture of the neural signal processor required for a given (signal) processor realization is therefore to be decided on the kind of measurements to be taken on the incident patterns (signals) and the type of decisions to be effected on the measurements. In this thesis, however, the specific aspect of (automated) procedures for guiding a selection of architecture is not attempted

The above discussion on the influence of the axioms of neural signal processing on the selection of the architecture of the neural signal processor is, in essence, an existential rephrasing of the constructive approaches suggested by **Geva & Sitte** (1992), **Zhang & Benveniste** (1992) and **Pati & Krishnaprasad** (1993). However, the implication that a type-2 neural signal processor with sigmoidal activation functions and linear discriminants is adequate for representing all functions realized by the entire class of (non-evolutionary) neural networks (and neural signal processors) described by the axioms of neural signal processing should not be missed. This implication has been the basis of the claims of **Hecht-Nielsen** (1987a), **Kůrková** (1992), **Kovačec & Ribeiro** (1993) and **Lagunas, Pérez-Neira**, *et al* (1993).

Recall Equation 5.14 (*p.* 300). In this equation, a type-2 feed-forward neural signal processor functionally equivalent to some neural signal processor (of the non-evolutionary kind) described by the axioms of neural signal processing, if the weights $\underline{w}_{i_j}$ are restricted in such a way that $\underline{w}_{i_j} = \underline{w}_i$, for an appropriate composition of the indexing variable i_j in terms of the index variables i and j, then the χ and π functions denote the following monotonic evaluations.

$$\chi_j(\,) = \beta_j \sigma_s(\cdot - \theta_j),$$

$$\pi_{i_j}(\,) = \alpha_{i_j} \sigma_s(\cdot - \theta_{i_j}),$$

$$\eta_{ij}(\underline{x}) \quad = \quad \underline{w}_{ij}\cdot\underline{x}\cdot$$

(Refer Theorem 5.4.1 with an appropriate change of variables.)

In the function representation scheme based on a solution, by Kolmogorov, of Hilbert's 13th problem, the π functions are required to be monotonic and independent of the family of functions being represented. Dependence on the functions represented is effected only through the χ functions. This implies that the representation of measurement functions is independent of the family of processors being realized. As a consequence, while the features, η, are not restricted to be independent of the specific processor being synthesized, the preliminary evaluation, or preprocessing, of the features is required to be independent of the family of processors synthesized. While this interpretation is not new in the domain of conventional signal processing, an implication of such an interpretation arising in the context of connectionist signal processing is that the axioms of neural signal processing under the earlier stated notion of representation (*ie*, function synthesis in a 'basis' which is itself synthesized as a sigmoidal transformation of the response of a neural signal processor) suggest the possibility of a mediation between the notions of (and approaches to) representation in traditional (symbolic) and connectionist approaches to AI.

Learning, in the context of neural signal processors that are interpreted as point-wise nonlinear transformations between integral transforms, is a process involving a specification of the kernels of the integral transforms. Though it is feasible to formulate learning as a search, through an appropriately stated gradient descent, for an appropriate kernel, such an approach is not considered in this thesis. Instead, an alternate approach is suggested wherein the kernel, recognized as a function on a suitable subspace of $\Re^{n_K}$, for an appropriate value of n_K, is synthesized through a neural signal processor. (A characterization of kernels $K(\xi, \gamma)$ as functions of two points was discovered by J. Mercer in 1909. See **Aronszajn**, 1950, for this historical aspect.)

Note that in a type-k neural signal processor, $k = 1, 2, \ldots$, the measurement kernels $K_w^{(\ell)}$ and $K_\epsilon^{(\ell)}$ are functions on $\Xi^{(\ell)} \times \Gamma^{(\ell-1)}$, where $\Xi^{(\ell)}$, $\ell = 1, 2, \ldots k$, is the index set for the collection of decision nodes in layer ℓ and $\Gamma^{(\ell)}$, $\ell = 0, 1, 2, \; ..k$, is the index set for the collection of concepts (inputs when $\ell = 0$), *ie* responses of the neural signal processor, in level ℓ. Similarly, the aggregation kernels $K_v^{(\ell)}$, $\ell = 1, 2, \ldots k$, are functions on $\Gamma^{(\ell)} \times \Xi^{(\ell)}$. When the index sets of the collection of processing nodes as well as the concepts are chosen to be one-dimensional and identified with a subset of $\Re$, then all the kernels are functions on appropriate subsets of $\Re^2$.

A synthesis, or design, of the kernels of integral transforms of a neural signal processor through an appropriately chosen neural signal processor necessitates examples of association between the domain and range of the kernels, *ie*, a training set that effectively is an *a priori* specification of the weights to be identified with some of the channels in the layer corresponding to the kernel under consideration. Such *a priori* information about some (or all) of the weights of a few of the processing nodes in each layer is available when the structure of the networked ensemble is partially known, as *eg*, in the case of hybrid neural networks. The approach of realizing kernels of integral transforms of a neural signal processor in the neural signal processing paradigm will be taken up in Chapter 6.

5.5 Summary

Neural signal processors, defined to be members of a typed class of abstract dynamical systems – the type number specifies the degree of association in the sense of layering – have been shown, recognizing the functional association to be time-indexed statements of spatial correlation in the incident input patterns, to represent continuous functions with arbitrary accuracy when the activation functions are continuous and non-constant. Denseness in representation is independent of the degree of association.

The principal focus in the study of the functional nature of neural signal processors has been to formulate the axioms relevant in neural signal processing: these axioms are listed below.

1. Axiom of Organization.

   A neural signal processor is composed of (layers of) three operational stages: measurement, discrimination and aggregation in that order. Preprocessing, if any, (preceding, or incorporated in, the measurement) is sought to be represented in a neural basis. Measurements are effected on an observation space constructed as the Cartesian product of the input space and a relevant subspace of a union of the space of responses of the distinct layers.

2. Axiom of Measurement.

   A neural signal processor, through the measurement functions in each of the processing (decision making) nodes, induces a foliation, of codimension at least one, in the input manifold. This foliation forms the basis of synthesizing (approximating) the desired level curves of the function.

3. Axiom of Discrimination

   A neural signal processor, through its discriminatory functions, renews the foliations, induced on the input space by the measurement functions, through a transformation, of the stems of the foliations, with at least one of the following properties:

   (a) alter the indexing of leaves to retain distinctness in a finite non-zero number of local regions of the input space,

   (b) introduce multiple components in the leaves,

   (c) associate, to at least one component of a leaf of the foliation due to discrimination, uncountably many leaves of the foliation due to measurement.

Re-foliations provide the basis for establishing equivalences between members (elements) of the input space in ways not possible through the chosen measurement functions.

4. Axiom of Aggregation

   A neural signal processor, through its aggregation function, synthesizes (or approximates) the level regions of processor response through a foliation on the Cartesian product of the stems of foliations on the input space due to discrimination. Concepts, in neural signal processors, are identified with the level regions of processor response.

It is important to appreciate that these axioms state the functional characteristics of the distinct components of neural signal processors and do not, in any sense, imply the specific details of the constituents of the processors: specification of the constituents, essential in the design of neural signal processors, will have to be addressed by the constraints imposed by the specific function representation problem at hand. Considering the interpretative scope of the term artificial neural networks (synonymous with connectionist information processing), discussed in § 1.1 (*p.* 5), the above axioms provide a pointer to the philosophical foundation of neural information processing.

Operationally, neural signal processors have been shown to effect (point-wise) nonlinear transformations between integral transforms: I suggest this operational character to be the representational paradigm of neural signal processing. Neural signal processors, in this interpretation, are compared with the conventional approach of realizing signal processors: the salient aspect of this comparison is that *function representation in neural signal processors is attempted by a process involving a search for kernels of integral transforms appropriate to the desired processor described through examples, the mechanism of association between the integral transforms being independent of the processor family, while the same is effected in conventional signal processing through an identification of an association appropriate to integral transforms evaluated independent of the family of processors.*

Features in the patterns presented to a processor, as understood in information processing contexts grounded in the current understanding of (human) perceptual abilities, are linked to integral transforms, the rationale being that kernels of integral transforms provide a template of the features, possibly known *a priori*, being discovered in input patterns. The representational paradigm of neural signal processing relates to the components described by the axiom of organization in the following sense. Measurements incorporate an extraction of features in the presented patterns, discrimination formalizes the decisions – in the nature of predicates of an appropriate mode of logic – taken on features and aggregation allows a synthesis of concepts through decisions.

Approximation being the methodological basis of function representation in neural signal processors, Kolmogorov's theorem on representation of multi-variate functions has been interpreted in the context of neural signal processing as being a characterization of the representational complexity of given concepts. The constituent functions of the representation in neural signal processors derived as an interpretation of Kolmogorov's theorem on function representation have been related to the kernels of aggregation, these kernels being in the class of kernels of nonlinear Urysohn operators. A few representational features of architectures based on the axioms of neural signal processing have been studied with the help of the function representation scheme suggested by an interpretation, in the context of neural networks, to Kolmogorov's theorem on function representation.

The representational nature of neural signal processors being one of inducing a foliation in the input space and the foliation, due to measurement, being effected through integral transforms it is of interest to know the nature of predicates operating on the input space. In particular, the possibility of localization in the predicates and, thereby, the concepts and the nature of localization in function representation is important in a study of neural signal processors. Localization in function representation becomes important as distinguishability between leaves is restricted to local regions by the axiom of discrimination. Chapter 6, the penultimate chapter of this thesis, will focus on the above mentioned issues of neural signal processing.

Chapter 6

Localization in Neural Signal Processing

It is common practice to use certain local operators to preprocess patterns prior to their recognition. The best known of these is the smoothing operator . . A neighbour set of numbers is summed and the sum compared with a .. threshold. If the sum is equal to or exceeds the threshold, a one is passed on to the next stage of processing ...., [else] a zero is passed on ... The process is repeated over the entire field of the pattern. This type of operator can be used to fill gaps in line patterns, to thicken lines, or to remove small irregularities. It is usual to take a symmetrical .. operator . [The] directional properties that it has stem from the intrinsic [symmetries in the operator].

— Michael J B Duff
Parallel Computation ın Pattern Recognıtıon,
in *Methodologies of Pattern Recognitıon,*
edited by Satosi Watanabe
Academic Press, New York, 1969

Neural information processing, as a paradigm of representation, is a statement of nonlinear association between integral transforms, as indicated in the previous chapter and the key departure of neural signal processing from classical approaches is in the incorporation of nonlinearity, in general, in the measurement and aggregation kernels, and in effecting a nonlinear transformation between the integral transforms related to measurement and aggregation: these notions, though, not unknown in the literature of signal processing, are not predominant due to analytical intractability. The integral transforms reflect the essence of signal representation and nonlinear association effected by activation functions is helpful in appraising the interplay, aided by layering, between signal and system[1] representation.

While in conventional signal processing, the kernels are chosen to be independent of the signal class under investigation and the focus is to study the nature of association required between integral transforms of the input and output signals in order to realize the desired processor functionality, the approach in neural information processing is to choose the kernel with the (point-wise) nonlinear association between integral transforms being independent of the processor class. Indeed, the kernels of the integral transforms of measurement and aggregation, referring, indirectly, to the strength of interconnections between processing nodes in the network, form the parameters through which the effective processing functionality is understood and synthesized.

[1] The reference is to information (signal) processing systems

An immediate consequence of this complementarity–the choice of association in conventional signal processing and that of kernels in neural signal processing–in operation between the two approaches to signal processing is that invertibility[2] of (integral) transforms and signal reconstruction which are of paramount importance in conventional schemes (based on transform domain techniques) do not find the same prominence in the neural paradigm. While these issues are not altogether irrelevant to neural information processing, at an operational level, the neural paradigm is not contingent on either invertibility of integral transforms, or the need for ensuring signal reconstruction.

The aggregation integral transforms are not restricted to be interpreted as the inverse of the integral transforms of measurement, and it is imperative to appreciate the vast degree of freedom in processor design ensuing as a consequence of this generalization. Though concepts represented at the outputs of neural signal processors belong to the same genre as the concepts (patterns or signals) presented as inputs or those in the intermediate levels of processing, as indicated while motivating the axiom of aggregation in § 5.2, it is important to note that the character of concepts, measured through representational complexity suggested by the interpretation to Kolmogorov's theorem in Theorem 5.4.1, is, in general, different at different levels of processing, thereby, providing ample scope for an exploitation of the representational freedom ensured by the aggregation kernel that differ

[2] This is a methodological requirement in conventional signal processing.

from the inverse of the measurement kernel· in studies of psychology, the complexity of concepts is considered to increase with the degree of association.

Despite these significant differences between the classical and neural approaches to signal processing, it should be noted that localization of evaluation, central to signal processing having connotations of feature extraction and evaluation of features, is still evident in neural signal processors. In this chapter, the nature, and influences of localization in neural signal processors will be studied. For simplicity in understanding, the discussion will begin by considering the sense in which isolated neurons effect a localization in their evaluation, and this study will be continued to appreciate the nature of localization in more general (layered) neural signal processors.

Characterization of localization and the implications of localization in the realization of signal processors with neural networks will form the focus of study in the latter part of the discussion. Predicates realized through neural signal processors are shown to have localized influence and through this have been related to window transforms: on the basis of the mechanism of localization, predicates are divided into 'intra-pattern' predicates and 'inter-pattern' predicates. Concepts are shown to be represented in neural signal processors as localized regions in the 'sheaf of input patterns.'

While the operational aspect of neural information processing is easily studied, in signal processing terminology, as nonlinear associations between integral transforms whereby the focus of processor representation reduces to a judicious selection of kernels for measurement and aggregation integral transforms (as the mechanism of nonlinear association is invariant to the processor class), the central problem of neural signal processing, inspired by processes of perceptual relevance, is considered to be the incorporation of knowledge of processing functionality available through examples of input-output association. Processor representation through kernel selection while implicitly, though indirectly, admitting 'learning by examples,' is more general, and accommodates the possibility of specifying kernels based on qualitative information about the class in which the processor belongs.

In § 6.1 I initiate a discussion on the nature of localization in isolated neurons wherein the weights (equivalents of kernels in isolated neurons), of processing nodes defined on function spaces, are established to be window functions in order that representation is non-trivial. § 6.2 (*p*. 322) is a study on the representation of localization in neural signal processors: localization related to the directional derivatives of activation functions that are window functions is in focus. A characterization of localization through window transforms and the implications of localization on processor representation have been studied in § 6.3 (*p*. 332). This chapter concludes with a cursory look, in § 6.4 (*p*. 344), into the influence of kernels on representation potential.

6.1 Nature of Localization in Isolated Neurons

Isolated neurons, imposing categorization on pattern spaces, are formally described as (see Chapter 2 for notations)

$$\begin{aligned} \dot{\eta}(\underline{x}, t) &= a(\eta(\underline{x}, t)) \, [b(\eta(\underline{x}, t)) + \underline{w} \; \underline{x}] \,, & (6.1a) \\ y(\underline{x}, t) &= \sigma(\eta(\underline{x}, t) - \theta) \,. & (6.1b) \end{aligned}$$

An extension of this formal model of neurons to incorporate decisions on function spaces is based on an alteration of the first of these expressions to the form (as indicated in § 5.3)

$$\dot{\eta}(x, t) = a(\eta(x, t)) \, [b(\eta(x, t)) + \langle w, x \rangle] \,, \tag{6.2}$$

with the added stipulation that w, and x are functions (on $\Re$);[3] η, a, b, and σ, while enjoying the same interpretation (and range spaces) as in the case of neural decision elements on Euclidean pattern spaces, are defined so as to be compatible for decision making on function spaces. The decision y is still a scalar, and decisions are taken on functions (x) that are possibly evolving in time (denoted by t), the dependence of x on t, however, is not explicitly indicated anywhere.

In the ensuing discussion, the only demand that will be made on the functional structure of σ is:

$$\lim_{\xi \to +\infty} \sigma(\xi) = \zeta_+, \qquad \lim_{\xi \to -\infty} \sigma(\xi) = \zeta_-,$$

[3] In the case of neural decision elements on function spaces, w and x belong to a Hilbert space of appropriate dimensions.

or, if $\lim_{\xi \to -\infty} \sigma(\xi) = \lim_{\xi \to +\infty} \sigma(\xi) = \zeta \in [\zeta_-, \zeta_+]$ then

$$\exists \xi \in (-\infty, +\infty) \text{ such that } \sigma(\xi) \neq \zeta.$$

This structure allows the decision space $\mathcal{Y}$ to be, minimally covered by, any compact, simply connected subspace of $\Re$: thus σ is unique up to isomorphisms (*eg*, linear, invertible transformations) mapping, say $[-1, 1]$, to $\mathcal{Y}$. For the sake of preciseness in argumentation, consider the following.

Definition **6.1.1** *A (decision) function $f. \mathcal{X} \to \mathcal{Y}$ is said to be trivial if it is a constant almost everywhere [ae], or if it is undefined ae.*

This definition, similar to the notion of trivial dichotomies defined in Chapter 3, allows the following characterization of function representation in (isolated) neurons: it is of interest to contrast the ensuing statement with the notion of preservance weights introduced in Chapter 3 (§ 3.1).

Proposition **6.1.1** *For any non-trivial (decision) function that is to have a representation in an isolated neuron, with $\theta \in \Re$,*

(a) the weight vector ($\underline{w}$) of the neuron (inducing decisions on (Euclidean) pattern spaces) should be in ℓ^2 (in general, in ℓ^p, $p = 2, 3, \ldots$),

(b) the weight function (w) of the neuron (inducing decisions on function spaces) should be in $L^2(\Re)$ (in general, in $L^p(\Re)$, $p = 2, 3,$).

𝒫ℛ𝒪𝒪ℱ:

(a) By definition, $\|\underline{w}\|_p^p = \sum_{i=1}^{n} |w^i|^p$, $p = 1, 2,$, and thus $\underline{w} \in \ell^p$ if $\forall i$ $|w_i| < \infty$, $i = 1, 2,$ $.n$ (n finite), *ie*, all elements of $\underline{w}$ are finite. If one or more elements of $\underline{w}$ are non-finite, then the innerproduct $\underline{w}\,\underline{x}$ is either $+\infty$, $-\infty$ or undefined, for all $\underline{x} \in \ell^p$ such that $\|\underline{x}\| \neq 0$ and $x_i \neq 0$ whenever $w_i \notin \Re$, $i = 1, 2, .$ n Since the number of points $\underline{x} \in \ell^p$ where $\underline{w}\cdot\underline{x}$ is neither $+\infty$, $-\infty$ nor undefined is atmost countable for finite n, $\underline{w}\,\underline{x} = +\infty, -\infty$ or undefined *ae*. For any $\theta \in \Re$, *ie*, $|\theta| < \infty$, $\underline{w}\,\underline{x} - \theta$ is either $+\infty$, $-\infty$ or undefined. If ϵ_w denotes the exception set, *ie*, $\epsilon_w = \{\underline{x} \in \ell^p \mid \underline{w}\cdot\underline{x}$ is neither $+\infty, -\infty$ nor undefined$\}$ then $\forall \underline{x} \in \ell^p \setminus \epsilon_w$ $\underline{w}\cdot\underline{x}$ and, hence, $\underline{w}\,\underline{x} - \theta$ equals c, where, c is either $+\infty$, $-\infty$ or undefined, which implies that $y(\underline{x}) = \mathfrak{c}$ for all $\underline{x} \in \ell^p \setminus \epsilon_w$, where, $\mathfrak{c}$ is either $+1$, -1 or undefined, noting that σ evaluated on an undefined domain point leads to an undefined range point. Thus if one or more elements w_i are non-finite, then the function represented is trivial, thereby refuting the negation of the statement.

(b) This component of the Proposition is similar to the finite dimensional case, (a), except that functions and the function space $L^p(\Re)$ in place of vectors (sequences) and the sequence space ℓ^p and, hence, a separate proof is not being provided.

□

The finite dimensional vector $\underline{x}$ and the function x are finite energy patterns, *ie*, $\underline{x} \in \ell^2$ and $x \in L^2(\Re)$, and the above Proposition stresses on $\underline{w}, \underline{x} \in \ell^2$ and $w, x \in L^2(\Re)$ to ensure that the innerproducts $\underline{w}\ \underline{x}$ and $\langle w, x\rangle$ are in $\Re$ (through Cauchy-Schwartz inequality), a necessary (but not sufficient) condition for the admissibility of representation of non-trivial functions in isolated neurons. I will now state a theorem from the theory of Fourier Transforms (see **Chui**, 1992 for a proof) which will be utilized occasionally. The operator of ordinary differentiation (of a function), with respect to the independent variable, is denoted by D.

THEOREM **6.1.1** *If $f \in L^1(\Re)$ then its Fourier transform $\hat{f}$ satisfies:*

1. *$\hat{f} \in L^\infty(\Re)$ with $\|\hat{f}\|_{L^\infty}(\Re) = \|\hat{f}\|_{L^1(\Re)}$,*
2. *$\hat{f}$ is uniformly continuous on $\Re$,*
3. *if the derivative Df of f also exists and is in $L^2(\Re)$ then*

$$(\widehat{Df})(\omega) = i\omega\hat{f}(\omega),\ i^2 = -1,$$

4. *$\hat{f}(\omega) \to 0$ as $\omega \to \pm\infty$ (Riemann-Lesbegue Lemma), and*
5. *if $f \in L^2(\Re)$, in addition, then $\hat{f} \in L^2(\Re)$ and $\|\hat{f}\|_{L^2(\Re)} = \|f\|_{L^2(\Re)}$ (Parseval identity).*

As a consequence of this theorem the following dual holds, the proof of which follows exactly on the lines of that for Theorem 6.1.1 and, hence, has not been included.

Proposition 6.1.2 *If $\hat{w} \in L^1(\Re)$ then the inverse Fourier transform w satisfies:*

1. *w is uniformly continuous on $\Re$,*
2. *if the derivative $D\hat{w}$ of $\hat{w}$ exists and $D\hat{w} \in L^1(\Re)$ then $Dw(\tau) = -\imath\tau w(\tau)$, and*
3. *$w(\tau) \to 0$ as $\tau \to \pm\infty$.*

The hypothesis $\hat{w} \in L^1(\Re)$ is a necessary condition for w to be reconstructed from $\hat{w}$ using the Fourier inversion rule:[4] thus $\hat{w} \in L^1(\Re)$ is assumed. If in addition[5] $\hat{w} \in L^2(\Re)$ then, through Parseval identity and the above Proposition, $w \in L^2(\Re)$, w is uniformly continuous on $\Re$, and $w(\tau) \to 0$ as $\tau \to \pm\infty$, *ie*, w is a *localized* function with vanishing asymptotes, which motivates the following.

Proposition 6.1.3 *If $w \in L^1(\Re) \cap L^2(\Re)$ and $\hat{w} \in L^1(\Re) \cap L^2(\Re)$, $D\hat{w}$ exists and is in $L^1(\Re) \cap L^2(\Re)$, then the function $\tau w(\tau) \in L^2(\Re)$.*

Proof: Under the assumed hypothesis, $\int\limits_{-\infty}^{+\infty} d\omega\, e^{\imath\omega\tau} D\hat{w} = -\imath\tau\hat{w}(\tau)$ from Theorem 6.1.1 (item 3) and by Parseval identity $||\tau w(\tau)||_{L^2(\Re)} = ||D\hat{w}||_{L^2(\Re)}$ which establishes the necessary statement.

□

[4] This feature is desired in all signal processing situations.

[5] The restriction of $\hat{x} \in L^1(\Re)$ and consequently of (uniform) continuity of x on $\Re$ is not intentionally imposed. However, the physical considerations of realization in patterns necessitates that $\hat{x} \in L^2(\Re)$

A function $w \in L^2(\Re)$ with the additional property that $rw(r) \in L^2(\Re)$ is known in the literature of Window transforms (*op cit*) as a *window function*, a term used to identify a class of localized functions with vanishing asymptotes. Noting that linear spans of (suitably chosen) window functions are dense in $L^2(\Re)$, this notion allows the preceding discussion to be summed up concisely as the following without the need for a proof.

THEOREM **6.1.2** *It is necessary that the weighting function (w) of an isolated neuron, defined to incorporate decisions on function spaces, be in a linear span of window functions in order that non-trivial (decision) functions are represented.*

As w is a linear combination of localized functions with vanishing asymptotes, the innerproduct $\langle w, x \rangle$, in some sense, allows one to consider a weighted average of restricted evaluations (discrimination) of the function x, the restriction is to some domain smaller than that over which x is defined. Owing to the continuity of w, the innerproduct $\langle w, x \rangle$ restricts attention to a connected subset of the domain of x. Precise characterization of localization of incident patterns x and connectedness of the local region of x will be considered in a later section.

The above theorem is applicable to dynamical, as well as, static versions of neurons defined to operate on function spaces. It is easy to see that in the case of dynamical neurons, localization of evaluation is really spatial as the evaluation of $\dot{\eta}$, and, consequently, that of η, at every point $t \in \Re_{0,+}$, is restricted to a (connected) region smaller than that over which x is defined. The region of localization is invariant in t (commonly having the connotations of time) as the weighting functions are defined to be indexed only by τ (spatial connotations) and not in t. Hence, temporal localization cannot be assured. All of the above observations can be easily transported to the realm of neurons defined on finite dimensional pattern spaces as admissibility of the weight vector $\underline{w}$ in ℓ^2 is assured by a weight $\underline{w}$ in the linear span of (suitably chosen) discrete window sequences.

6.2 Representation of Localization in Neural Signal Processors

Since processing in isolated neurons defined on function spaces is localized, it is natural to seek the nature of representation in neural signal processors described by the functional form

$$\eta^{(\ell)}_{\xi^{(\ell)}}(x,t) = \langle \boldsymbol{K}^{(\ell)}_{w}(\xi^{(\ell)},\cdot), \mathfrak{y}^{(\ell-1)}(x,t) \rangle - \theta^{(\ell)}_{\xi^{(\ell)}}, \tag{6.3a}$$

$$y^{(\ell)}_{\xi^{(\ell)}}(x,t) = \sigma^{(\ell)}_{\xi^{(\ell)}}(\eta^{(\ell)}_{\xi^{(\ell)}}(x,t)), \tag{6.3b}$$

$$\mathfrak{y}^{(\ell)}_{\gamma^{(\ell)}}(x,t) = \langle \boldsymbol{K}^{(\ell)}_{v}(\gamma^{(\ell)},\cdot), y^{(\ell)}(x,t) \rangle - \theta^{(\ell)}_{\gamma^{(\ell)}} \tag{6.3c}$$

where, $x \in \mathfrak{X}$, $y \in \mathfrak{Y}$, η takes values in $\Re$, $\mathfrak{y}^{(0)}(x,t) \equiv x \ \forall x \in \mathfrak{X}$, $\forall t \in \Re_{0,+}$, and $\xi^{(\ell)} \in \Xi^{(\ell)}$, $\ell = 1,2,\ldots,k$, $\gamma^{(\ell)} \in \Gamma^{(\ell)}$, $\ell = 0,1,2,\ldots,k$, $\Xi^{(\ell)}$, and $\Gamma^{(\ell)}$, for appropriate ℓ, are allowed to be either continuous, or discrete, depending on the requirements of processing and/or analysis. With an abuse of notation the following (simplified) convention is used.

$$\langle K_w^{(\ell)}(\xi^{(\ell)},\gamma^{(\ell-1)}),\mathfrak{y}_{\gamma^{(\ell-1)}}^{(\ell)}(x,t)\rangle = \begin{cases} \displaystyle\int_{\Gamma^{(\ell-1)}} d\mu(\gamma^{(\ell-1)})\, K_w^{(\ell)}(\xi^{(\ell)},\gamma^{(\ell-1)})\mathfrak{y}_{\gamma^{(\ell-1)}}^{(\ell-1)}(x,t) \\ \qquad \text{if } \Gamma^{(\ell-1)} \text{ is a continuous space,} \\ \displaystyle\sum_{\gamma^{(\ell-1)}\in\Gamma^{(\ell-1)}} K_w^{(\ell)}(\xi^{(\ell)},\gamma^{(\ell-1)})\mathfrak{y}_{\gamma^{(\ell-1)}}^{(\ell-1)}(x,t) \\ \qquad \text{if } \Gamma^{(\ell-1)} \text{ is discrete} \end{cases}$$

The other two inner product species are also similarly denoted. Proposition 6.1.1 (*p*. 317) immediately implies the following.

𝔓𝔯𝔬𝔭𝔬𝔰𝔦𝔱𝔦𝔬𝔫 **6.2.1** *A necessary condition for the functions $\mathfrak{y}_{\gamma^{(\ell)}}^{(\ell)}$, $\forall\gamma^{(\ell)} \in \Gamma^{(\ell)}$, $\ell = 1,2,\ldots k$, to be non-trivial is that the measurement weighting functions, $w_{\xi^{(\ell)}}^{(\ell)}$ and $\epsilon_{\xi^{(\ell)}}^{(\ell)}$, in type-k neural signal processors, ${}^k\mathfrak{N}(t)$, for all $t \in \Re_{0,+}$, are in a linear combination of window functions, for $k = 1,2,\ldots$, $\forall\xi^{(\ell)} \in \Xi^{(\ell)}$, $\ell = 1,2, \ldots k$. In addition, it is necessary that the aggregation weighting functions, $v_{\gamma^{(\ell)}}^{(\ell)}$, be in a linear combination of window functions to ensure that the responses, $\mathfrak{y}_{\gamma^{(\ell)}}^{(\ell)}$ (with a quantification as indicated earlier), of the neural signal processor be bounded.*

Noting the way kernels have been defined in § 5.3, the preceding statement implies that kernels of the measurement and aggregation integral transforms are described by the following functional forms.

$$K_w^{(\ell)}(\xi^{(\ell)},\gamma^{(\ell-1)}) = \sum_{\imath\in\mathcal{I}_w} \left(\varpi_{w_\imath}^{(\ell)}(\bar{\xi}_{w_\imath}^{(\ell)}(\xi^{(\ell)}),\bar{\theta}_{w_\imath}^{(\ell)}(\xi^{(\ell)}))\right)(\gamma^{(\ell-1)}) \qquad \forall\gamma^{(\ell-1)}\in\Gamma^{(\ell-1)}, \tag{6.4a}$$

$$K_{\epsilon}^{(\ell)}(\xi^{(\ell)},\gamma^{(\ell)}) = \sum_{\imath\in\mathcal{I}_{\epsilon}} \left(\mathfrak{w}_{\epsilon_\imath}^{(\ell)}(\bar{\xi}_{\epsilon_\imath}^{(\ell)}(\xi^{(\ell)}),\bar{\theta}_{\epsilon_\imath}^{(\ell)}(\xi^{(\ell)}))\right)(\gamma^{(\ell)}) \quad \forall\gamma^{(\ell)}\in\Gamma^{(\ell)}, \tag{6 4b}$$

$$K_{\upsilon}^{(\ell)}(\gamma^{(\ell)},\xi^{(\ell)}) = \sum_{\imath\in\mathcal{I}_{\upsilon}} \left(\mathfrak{w}_{\upsilon_\imath}^{(\ell)}(\bar{\xi}_{\upsilon_\imath}^{(\ell)}(\gamma^{(\ell)}),\bar{\theta}_{\upsilon_\imath}^{(\ell)}(\gamma^{(\ell)}))\right)(\xi^{(\ell)}) \quad \forall\xi^{(\ell)}\in\Xi^{(\ell)} \tag{6.4c}$$

In these expressions, $\mathcal{I}$, with an appropriate suffix, denotes the specific index values participating in the linear combinations, $\mathfrak{w}$ is used to denote a window function (as indicated in § 2.1), and the dependence of spatial and spectral localization on node indices (ξ and γ) is abstracted by the functions $\tilde{\xi}$ and $\tilde{\theta}$, respectively. Relationships between structural aspects of the kernel and the representational character of neural signal processors will be focused in § 6.4 (p. 344).

According to the above Proposition, $\eta_{\xi^{(\ell)}}^{(\ell)}$, the measurement pertaining to the node indexed by $\xi^{(\ell)} \in \Xi^{(\ell)}$ in layer ℓ, is a weighted average of localized evaluations of the pattern (of activity) indicated by the concepts (or responses) $\mathfrak{y}^{(\ell-1)}$ of the preceding layer (subject to the understanding that $\mathfrak{y}^{(0)} \equiv x$) and $\mathfrak{y}^{(\ell)}$ of the same layer, but the previous time step. Responses of the type-k neural signal processor $\mathfrak{y}_{\gamma^{(\ell)}}^{(\ell)}$, at node $\gamma^{(\ell)} \in \Gamma^{(\ell)}$ in layer ℓ, is also indicated to be a localized evaluation of the decisions (inferences) $y^{(\ell)}$ of the corresponding k-layered neural network which is the substrate of the type-k neural signal processor.

The outputs $y^{(\ell)}$ are the result of the activation functions $\sigma^{(\ell)}$ acting (point-wise) on $\eta^{(\ell)}$, and it is of interest to investigate the role of this nonlinear (discriminatory) processing stage in the representation of localization in neural signal processors. Note that the activation functions $\sigma^{(\ell)}_{\xi^{(\ell)}}$, $\forall \xi^{(\ell)} \in \Xi^{(\ell)}$, $\ell = 1, 2, . . , k$, of a processor in ${}^{k}\mathfrak{N}(t)$ are, in general, functions with alternate stretches of monotonicity and smoothness (at least) at the (possibly vanishing) asymptotes, typical examples being the sigmoidal functions and radial basis (Gaussian) functions.

Gaussian functions, as described in § 2.2, are, by definition, window functions, and hence discrimination effected by such functions are of a localized nature: this feature, together with norm based discriminants has been considered with sufficient interest in radial basis function networks in view of the advantages, especially in parameter specification (*ie*, learning), offered by concepts that essentially reflect localized (possibly compact) regions in the input pattern space. It is quite disheartening to note that sigmoidal activation functions are not in $L^2(\Re)$, and, consequently, these functions are not window functions, and the lack of assurance in localization of evaluation has prompted radial basis function networks to be considered superior to those employing sigmoidal discrimination of linear measurements.

However, monotonic activation functions, *eg*, sigmoidal functions, induce localization in the synthesized processor in a sense described in the following. Though considerations of denseness in representation restrict the activation functions σ to be different from algebraic polynomials [*ae*] (*cf*, **Leshno**, **Ya Lin**, *et al*, 1994), it should be noted that the ensuing discussion on localization applies, equally, to all kinds of monotonicity in σ.

PROPOSITION **6.2.2** *$D^j\sigma$, $j = 1, 2, ..$, the derivative of a continuous monotonic function σ, if it exists, with smoothness at the asymptotes is a window function.*

PROOF: This statement will be established through the principle of mathematical induction on the order of differentiation.[6]

Verification

Monotonicity implies that the first derivative, if it exists, is one-sided. The additional requirement of smoothness at the asymptotes, *ie*, all derivatives of σ that exist should vanish at $\pm\infty$, together with continuity immediately establishes that $D\sigma$ is a window function.

Inference

Consider that $D^j\sigma$, for some j, $j = 1, 2, ..$, is a window function. In view of the hypothesis that σ is smooth at the asymptotes it immediately follows that $\widehat{D^j\sigma}$, the Fourier transform of $D^j\sigma$, is also a window function. As $D^{j+1}\sigma$, the derivative of $D^j\sigma$, is described in the spectral domain as $\widehat{D^{j+1}\sigma}(\omega) = \frac{\widehat{D^j\sigma}(\omega)}{\omega}$, it immediately follows that $\widehat{D^{j+1}\sigma}$ is in $L^2(\Re^n)$. This implies, by Parseval identity, that $D^{j+1}\sigma$ is in $L^2(\Re^n)$. Smoothness of σ at the asymptotes implies that $D^{j+1}\sigma$, the derivative of $D^j\sigma$, is a window function.

[6] D denotes the operator of differentiation with respect to the independent variable.

Conclusion

The validity of the claim for $j = 1$ and the assurance of validity for $j + 1$ conditional on the validity for every values of j, $j = 1, 2, \ldots$, establishes the stated claim.

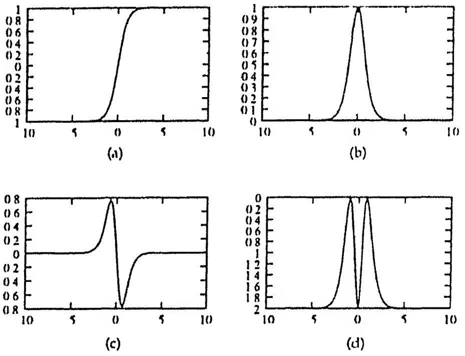
□

Figure 6.1 illustrates the nature of localization in the first three derivatives of a sigmoidal activation function. Localization of this nature is not restricted to monotonic functions alone and is applicable to activation functions that are continuous and piece-wise monotonic.[7] As the linear span of sigmoidal functions (in fact, continuous monotonic functions that are different from algebraic polynomials) is dense in the space of continuous functions (as demonstrated by **Cybenko**, 1989, and in § 5.1) and the differentiation operator is linear, the following statement is obvious.

PROPOSITION **6.2.3** *$D^j\sigma$, $j = 1, 2, \ldots$, the derivative of a continuous piece-wise monotonic function σ, if it exists, with smoothness at the asymptotes, is a window function.*

The above statements assure that localization in discrimination, with piece-wise monotonic functions which are smooth at the asymptotes, is available, if not in σ, at least in the variations of σ. Further,

[7] The derivatives of the sigmoidal activation function are good examples of piece-wise monotonic functions.



(a) (b) (c) (d)

Figure 6.1: Derivatives of a sigmoidal activation function. (a) $\sigma(\xi) = tanh(\xi)$, (b) $(D\sigma)(\xi) = 1-\sigma^2(\xi)$, (c) $(D^2\sigma)(\xi) = 2\sigma^3(\xi) - 2\sigma^2(\xi)$, (d) $(D^3\sigma)(\xi) = 8\sigma^2(\xi) - 8\sigma^4(\xi) - 2, \xi \in \Re$

localization effected by piece-wise monotonic functions is, in general, a weighted average of local functions in the sense that the derivatives of such functions belong to linear spans of window functions obtained as derivatives of the sigmoidal activation function. It would not be inappropriate to interpret discrimination as an estimation of the localization in evaluation due to the specific realization of the activation function σ from sigmoidal functions.

While the influence of σ on the measurements η is to cause the neural action (decision) y to depend only on a local range of η, it is imperative that the effect of this localized evaluation be known over the input pattern space $\mathfrak{X}$ (a function space). Noting that the input x

is, in this mildly generalized discussion, a function, attention will be restricted to a linear subspace[8]

$$\mathcal{L}_{x_a} = \{x | x = \alpha x_a,\ \alpha \in \Re\},\ x_a \in \mathfrak{X},\ \|x_a\|_{L^2(\Re)} = 1.$$

Since $\mathcal{L}_{x_a}$ is the collection of range scaled versions of x_a, generality is not lost in assuming $\|x_a\|_{L^2(\Re)} = 1$. To aid the study of localization, I will consider the notion of directional derivatives (**Corwin & Szczarba**, 1982): this notion is reproduced below.

If f denotes a function defined over every element of $\mathcal{L}_{x_a}$, then the *directional derivative*[9] operator (also termed as directional differentiation operator) D_{x_a}, subject to the requirements of ordinary differentiation on f, is given by

$$(D_{x_a})(x) = \lim_{\delta \to 0} \frac{f((\alpha + \delta)x_a) - f(\alpha x_a)}{\alpha}, \quad x = \alpha x_a.$$

This operator allows a consideration of the evolution of f relative to x in the *direction* x_a, the evolution being evaluated at the *point* x. When the dimensionality of $\mathfrak{X}$ is unity, as in the case of the range space of inner products (discrete as well as continuous), then the directional derivative operator reduces to the familiar ordinary differentiation operator. The directional differentiation operator has characteristics similar to the ordinary differentiation operator, in particular, the following.

[8] The symbol α is being reused in this chapter to mean the scale factor associated with x The earlier connotations of α being the norm of a (preservance) weight and index of leaves in a foliation are being discontinued.

[9] The notion of directional derivative is to be considered in the context of functions defined over finite-dimensional as well as infinite-dimensional spaces

1. *Additivity*: $(\boldsymbol{D}_{x_a}(f+g))(x) = (\boldsymbol{D}_{x_a}f)(x) + (\boldsymbol{D}_{x_a}g)(x)$, $\forall f, g \in \mathfrak{X}$,

2. *Homogeneity*: $(\boldsymbol{D}_{x_a}\beta f)(x) = \beta(\boldsymbol{D}_{x_a}f)(x)$, for all $f \in \mathfrak{X}$, $\beta \in \Re$ and invariant with respect to x.

3. *Product Rule*: $(\boldsymbol{D}_{x_a}(fg))(x) = ((\boldsymbol{D}_{x_a}f)g)(x) + (f(\boldsymbol{D}_{x_a}g))(x)$, for all $f, g \in \mathfrak{X}$,

4. *Chain Rule*:[10] $(\boldsymbol{D}_{x_a}f(g))(x) = ((\boldsymbol{D}_f)(\boldsymbol{D}_{x_a}g)(x)$ for all $f, g \in \mathfrak{X}$,

for all $x \in \mathcal{L}_{x_a}$. The $\jmath$th order directional derivative operator in the direction x_a will be denoted by $\boldsymbol{D}^{\jmath}_{x_a}$, $\jmath = 1, 2,$, the superscript denoting the repetitive application of the first-order directional derivative operator in the direction x_a, *ie*, $\boldsymbol{D}_{x_a}$.

PROPOSITION **6.2.4** *The $\jmath$th order directional derivative, $j = 1, 2,$. , in the direction x_a of the response $\mathfrak{y}^{(1)}_{\gamma^{(1)}}$, $\gamma^{(1)} \in \Gamma^{(1)}$, of a type-1 neural signal processor in ${}^1\mathfrak{N}(t)$, for all $t \in \Re_{0,+}$, restricted to $\mathcal{L}_{x_a} \subset \mathfrak{X}$, is given, as a function of $\alpha \triangleq \frac{\|x\|}{\|x_a\|}$, by*

$$\begin{aligned}(\boldsymbol{D}^{\jmath}_{x_a}\mathfrak{y}^{(1)}_{\gamma^{(1)}}(x,t))(\alpha) \quad = \quad & \Big\langle \boldsymbol{K}^{(1)}_v(\gamma^{(1)}, \xi^{(1)}), \\ & ((\boldsymbol{K}^{(1)}_w(\xi^{(1)}, \ ), x_a(\cdot)))^{\jmath} D\sigma^{(1)}_{\xi^{(1)}}(\alpha\langle \boldsymbol{K}^{(1)}_w(\xi^{(1)}, \ ), x_a(\cdot)\rangle - \theta^{(1)}_{\xi^{(1)}})\Big\rangle . \end{aligned} \tag{6.5}$$

[10] The operator D_f denotes ordinary differentiation with respect to the function f

$\mathcal{PROOF}$: From Equation 6.3 (p. 322) and the properties of the directional derivative operator, it is simple to see the following.

$$(D^{\jmath}_{x_a}\mathfrak{y}^{(1)}_{\gamma^{(1)}}(x,t))(\alpha) \quad = \quad \langle K^{(1)}_{v}(\gamma^{(1)},\xi^{(1)}),(D^{\jmath}_{x_a}y^{(1)}_{\xi^{(1)}}(x,t))\rangle \tag{6.6}$$

$$(D^{\jmath}_{x_a}\eta^{(1)}_{\xi^{(1)}}(x,t))(\alpha) \quad = \quad \begin{cases} \langle K^{(1)}_{w}(\xi^{(1)},\gamma^{0}),x_a\rangle & \text{if } \jmath = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{6.7}$$

The required result is obtained by applying chain rule of directional derivative operators while evaluating $(D^{\jmath}_{x_a}y^{(1)}_{\xi^{(1)}}(x,t))(\alpha)$, the directional derivatives of the neural decisions in the first layer.

□

Proposition 6.2.2 (p. 326) and Proposition 6.2.3 (p. 327) establish that the derivative $D\sigma$ of an activation function σ is a local function of its argument when the activation function is chosen to be piece-wise monotonic, *ie*, the class of functions satisfying the axiom of discrimination. For the same class of activation functions, it follows that the directional derivatives of the response of a type-1 neural signal processor is a weighted average of window functions: the weighting values are provided by the kernel of aggregation. Proposition 5.2.3 (p. 252) indicates that the functionality of a type-k neural signal processor, expressed as an indexed collection of operators on $\mathfrak{X}$ – the index space is $\Re_{0,+}$ – is composed of k operators, each representing the functionality of an appropriate type-1 neural signal processor. These observations lead to the following statement.

THEOREM **6.2.1** *The jth order directional derivatives, $j = 1, 2,$ . . , of the response of a type-k neural signal processor, $k = 1, 2,$, belong to the linear span of window functions.*

PROOF: When $k = 1$, the above statement is equivalent to Proposition 6.2.4. For other values of k the statement follows from an application of the chain rule for directional derivative operators on a k-stage decomposition equivalent to the type-k neural signal processor (*cf*, Proposition 5.2.3 (*p*. 252).)

□

6.3 Characterization of Localization

Localization of processor functionality, as studied in the preceding sections, is essentially a spatial characterization. From a signal processing point of view, the localization need not be restricted to spatial characterizations and can also include spectral characterizations. Thus it is not adequate only to investigate spatial localizations. Since the spatial characterization shows that the influence of kernels of the integral transforms related to measurement and aggregation in neural signal processors is one of restricting the weighting functions (weights) to be in the linear span of window functions (sequences) and the directional derivatives of neural signal processor response are similarly localized functions, an analysis based on principle of uncertainty, typical in discussions involving window transforms,[11] would provide the necessary links between spatial and spectral localizations.

In this section, I will begin with a characterization of the localization of an isolated neuron preparatory to a characterization of the influence of kernels on the representation provided by type-1 neural signal processors. Following this I will investigate the nature of spatial-spectral localization induced in neural signal processor response by the activation functions. The weighting function $w \in \mathfrak{W}$ in an isolated neuron having been established, in § 6.1 (*p*. 316), as a spatially localized function with vanishing asymptotes, a spectral characterization is provided through the following.

PROPOSITION **6.3.1** *If the function w is such that $\hat{w} \in L^1(\Re) \cap L^2(\Re)$, and the first j derivatives of $\hat{w}$ exist then*

1. *if the first j derivatives of $\hat{w} \in L^1(\Re) \cap L^2(\Re)$ then $\tau^p w(\tau) \in L^2(\Re)$, $p = 1, 2, \ldots, j$.*
2. *if the first j derivatives of $\hat{w}$ are zero at the origin (*ie*, $\omega = 0$, ω denoting the spectral indexing variable) then $\int_{-\infty}^{+\infty} d\tau\, \tau^p w(\tau) = 0$, $p = 1, 2, . \, . j$,* ie, *the first j moments vanish.*

This Proposition is stronger than Proposition 6.1.3 (*p*. 320), and suggests the correlation between the (Fourier) spectral nature of the

[11] See Chapter 2 for the essential aspects of window transforms.

weighting function w and the potential of w being a window function with vanishing moments. From signal processing considerations $\acute{w} \in L^1(\Re) \cap L^2(\Re)$ is not an unrealistic assumption and, hence, the above property of weighting functions $w \in \mathfrak{W}$ will be assumed

Window functions with at least the zeroth moment (essentially the average value) vanishing, *ie*, $\int_{-\infty}^{+\infty} d\tau\, w(\tau) = 0$, are considered as *basic wavelet windows* in the theory of wavelets, and considerations of *computational conveniences* and *stability* require as many, as possible, of the initial moments to vanish However, the requirement of vanishing moments would severely restrict the choice of weighting functions, and would disallow many valid candidate functions: the Gaussian function, a typical example of window function, with non-zero mean value has none of its moments vanishing.

Yet, a consideration of weighting functions in terms of wavelets is attractive enough not to be missed, and as an inevitable compromise between the conflicting requirements, the weighting functions will be considered as a mixture of wavelets, typically affine combinations familiar in the wavelet representation of functions However, such affine mixtures, by virtue of linearity, will still have its initial moments vanishing as indicated in the following: this is true only when the component wavelets in the mixture are not subjected to a null scale value. (A scaling by a null value, *ie* 0, is operationally the equivalent of introducing a constant.)

PROPOSITION 6.3.2 *Define the pth moment of a function f, $f \cdot \Re \rightarrow \Re$, as $M^p_f(\tau) \triangleq \int_{-\infty}^{+\infty} d\tau\, \tau^p f(\tau)$. Given two basic wavelets $\mathfrak{b}_1$ and $\mathfrak{b}_2$ such that their first j moments vanish,* ie, *$M^p_{\mathfrak{b}_i}(\tau) = 0$, $p = 1, 2, \ldots j$, $i = 1, 2$. Let $w(\tau) = \alpha_1 \mathfrak{b}_1(\beta_1 \tau - \theta_1) + \alpha_2 \mathfrak{b}_2(\beta_2 \tau - \theta_2)$, $\beta_1, \beta_2 \neq 0$, $\alpha_1, \alpha_1, \theta_1, \theta_2 \in \Re$. Then the moments of w vanish for $p = 0, 1, \ldots j$, as*

$$M^p_w(\tau) = \frac{\alpha_1}{\beta_1} \sum_{i=0}^{p} \binom{p}{i} (-\theta_1)^{p-i} M^i_{\mathfrak{b}_1}(\tau) + \frac{\alpha_2}{\beta_2} \sum_{j=0}^{p} \binom{p}{j} (-\theta_2)^{p-j} M^j_{\mathfrak{b}_2}(\tau) .$$

Association of the weighting function w with a window function enables a characterization of the nature of spatial localization induced in the response of a neural signal processor. The following statements are based on arguments in **Chui** (1992).

PROPOSITION 6.3.3 *A weighting function w derived from a basic wavelet window $\mathfrak{b}$ by scaling and/or translation*

$$w(\tau) = \mathfrak{b}(\beta\tau - \xi), \ \forall \tau \in \Re, \ \beta \neq 0, \ \zeta \in \Re, \tag{6.8}$$

where β is the scale factor and ξ is the translation, localizes evaluation of the innerproduct $\langle w, x \rangle$ to the index window (ie *spatial localization)*

$$\mathfrak{I}_w \triangleq \left[\xi + \frac{\tau^*_{\mathfrak{b}}}{\beta} - \frac{\Delta_{\mathfrak{b}}}{\beta}, \xi + \frac{\tau^*_{\mathfrak{b}}}{\beta} + \frac{\Delta_{\mathfrak{b}}}{\beta} \right]$$

of the input pattern (function) x. The index window is centered at $\tau^*_w = \xi + \frac{\tau^*_{\mathfrak{b}}}{\beta}$ and has a width $\Delta_w = 2\frac{\Delta_{\mathfrak{b}}}{\beta}$, where, $\tau^*_{\mathfrak{b}}$ and $\Delta_{\mathfrak{b}}$ are, respectively, the center and width of the basic wavelet $\mathfrak{b}$.

𝒫ℛ𝒪𝒫𝒪𝒮ℐ𝒯ℐ𝒪𝒩 **6.3.4** *The index window* $\mathfrak{I}_w$ *of a weighting function* w *given by* Equation 6.8 *is a connected subset of* $\Re$ *if the basic wavelet* $\mathfrak{b}$ *is a continuous function.*

𝒯ℋℰ𝒪ℛℰℳ **6.3.1** *Weighting functions* $w \in \mathfrak{W} \subseteq L^1(\Re) \cap L^2(\Re)$ *having a wavelet representation*

$$w(\tau) = \sum_{\imath \in \mathcal{I}} \alpha_\imath \mathfrak{b}_\imath(\beta_\imath \tau - \xi_\imath)$$

where $\{\mathfrak{b}_\imath\}_{\imath \in \mathcal{I}}$ *is a collection of basic wavelets, and* $\alpha_\imath = \langle w, \mathfrak{b}_\imath \rangle$, $\imath \in \mathcal{I}$, *exhibit the following.*

1. *The innerproduct* $\langle w, x \rangle$ *of the input pattern (function)* x *with the* w *evaluates* x *on a localized region of its definition.*

2. *An isolated neuron equipped with this weighting function* w *and discrimination enforced by the activation function* σ *evaluates a predicate that is a weighted average of localizations on the input pattern (function)* x*; this predicate is essentially an assessment of relative organization of assignments in* x *and averaging of constituent localized weightings of* x*, decided by the coefficients of representation of* w *in the collection of wavelets* $\{\mathfrak{b}_\imath\}_{\imath \in \mathcal{I}}$*, is over different scales and shifts.*

Piece-wise monotonicity in activation functions, a feature typical of the prevailing tradition in neural signal processing, has been shown in

§ 6.1 to incorporate localization in the response of an isolated neuron. Localization due to monotonicity in activation functions that are also continuous, and, preferably, analytic is in the sense of the directional derivatives of the neural response, expressed as a function of traversal in the direction of differentiation, being window functions.

Isolated neurons whose activation functions are represented as a linear combination of sigmoidal functions, spanning[12] the class of neurons with piece-wise monotonic activation functions, are similar in structure and function to neural signal processors, and consequently, the characterization of the nature of localization in the predicates of isolated neurons is provided later on in this section through a characterization of the localization in the predicates of neural signal processors. Presently I will focus on the nature of processors represented in isolated neurons due to sigmoidal activation functions and will hint at the similarity in representation with sigmoidal and Gaussian activation functions.

PROPOSITION **6.3.5** *For a sigmoid function*

$$\sigma(\xi) = \sigma_s(\xi) \triangleq (\zeta_+ - \zeta_-)\frac{(1 + tanh(\xi))}{2} + \zeta_-$$

($\sigma \in C^\infty(\Re)$), the derivatives $D^j\sigma$, $j = 2, 3, \ldots$, are

1. *basic wavelet windows,* ie, $\int_{-\infty}^{+\infty} d\xi\, D^j\sigma(\xi) = 0$,

[12] Recall the density theorem, originally, due to **Cybenko** (1989), and rephrased in § 5.1. This theorem assures that any continuous function of interest can be represented, with arbitrary accuracy, through finite linear combinations of sigmoidal functions

2. *admissible wavelets,* ie, $C_{D^j\sigma} = 2\pi \int_{-\infty}^{+\infty} d\omega\, |\omega|^{-1} |\widehat{D^j\sigma}(\omega)|^2 < \infty$.

𝒫ℛ𝒪𝒪ℱ: The first part is obvious from the definition of the sigmoid function. Figure 6.1 (*p.* 328) provides ample justification for the statement. I will prove only the admissibility of the derivatives of sigmoids as wavelets.

Denote $\hat{f}_j(\omega) = \frac{1}{\omega}\widehat{D^j\sigma}(\omega)$. This simplifies the expression for $C_{D^j\sigma}$ to

$$\frac{1}{2\pi} C_{D^j\sigma} = \int_{-\infty}^{+\infty} |\hat{f}_j(\omega)|\, |\widehat{D^j\sigma}(\omega)|\,.$$

By Cauchy-Schwartz inequality,

$$\frac{1}{2\pi} C_{D^j\sigma} \leq \|\hat{f}_j(\omega)\|_{L^2(\Re)} \, \|\widehat{D^j\sigma}(\omega)\|_{L^2(\Re)} = \|f_j(\xi)\|_{L^2(\Re)} \, \|\widehat{D^j\sigma}(\xi)\|_{L^2(\Re)}\,.$$

The right side in the above expression is simply Parseval's identity. Thus $C_{D^j\sigma} < \infty$ if $f \in L^2(\Re)$ and $D^j\sigma \in L^2(\Re)$. As $\hat{f}_j(\omega) = \frac{1}{\omega}\widehat{D^j\sigma}(\omega)$, it is simple to observe from the theory of Fourier transforms that $f_j(\xi) = \int_{-\infty}^{\xi} d\tau D^j\sigma(\tau)$, *ie,* f_j is the integral of $D^j\sigma$. From the definition of the sigmoid function, the desired result is established for $j = 2, 3, \ldots$, noting that for these values of j, the local nature of the derivatives of σ (established in Proposition 6.2.2 (*p.* 326) and Proposition 6.2.3 (*p.* 327)) ensures the existence of f_j and, thereby, $C_{D^j\sigma}$.

□

𝒫ℛ𝒪𝒫𝒪𝒮ℐ𝒯ℐ𝒪𝒩 **6.3.6** *A Gaussian function*

$$\sigma(\xi) = \sigma_g(\xi) \triangleq (\zeta_+ - \zeta_-) exp(-\frac{\xi^2}{2}) + \zeta_-,$$

easily seen to be an analytic function, has its derivatives $D^j\sigma$, $j = 1, 2, \ldots$, *as*

1. *basic wavelet windows,* ie, $\int_{-\infty}^{+\infty} d\xi\, D^j\sigma(\xi) = 0$,
2. *admissible wavelets,* ie, $C_{D^j\sigma} = 2\pi \int_{-\infty}^{+\infty} d\omega\, |\omega|^{-1} |\widehat{D^j\sigma}|^2 < \infty$.

The above result is not surprising on observing that the derivative of a sigmoidal function, $D\sigma_s(\xi) = \frac{1}{2}(\zeta_+ - \zeta_-)sech^2(\xi)$, a window function, is a very good approximation to the Gaussian function σ_g, except for range translation. As a consequence of the above Propositions, the following is evident.

PROPOSITION **6.3.7** *The directional derivatives of the response of an isolated neuron, in any direction, expressed as a function of traversal in the direction of differentiation, is a basic wavelet window, and is an admissible wavelet.*

Concepts represented by neural signal processors are local in character, and the nature of localization is strongly dependent on choices made regarding weighting functions (*ie*, measurement and aggregation kernels) and activation functions. Recall the operational nature[13] of a simplified version of an isolated neuron,

$$y(x) = \sigma(\langle w, x\rangle - \theta),$$

[13] This reduces to Equation 5.7 (*p* 276) if x and, consequently, w are drawn from a finite-dimensional Hilbert space, a situation reflected in the notations $\underline{x}$ and $\underline{w}$, respectively, with the understanding that $\langle \underline{w}, \underline{x}\rangle \equiv \underline{w}\cdot\underline{x}$.

where $\theta \in \Re$ is the threshold and

$$w(\xi) = \sum_{i=1}^{N} (\mathfrak{w}_i(\tilde{\xi}_i, \tilde{\theta}_i))(\xi), \forall \xi \in \Xi,$$

where $\mathfrak{w}$ is a window function whose spatial and spectral characteristics are determined by $\tilde{\xi}$ and $\tilde{\theta}$, N is *a priori* chosen, and Ξ is the common space[14] of definition of x and w. It is not difficult to visualize that the neural action (decision) is a statement comparing a weighted *average* of localized assessments of x with the threshold θ, the sense of averaging and localization being decided by the specific window functions in use.

With the binary comparator $\sigma = \sigma_h$ (see § 2.2), it would not be erroneous to declare that an isolated neuron evaluates a *predicate* on the input pattern x ($\underline{x}$), the nature of the predicate is decided by the component window functions of w ($\underline{w}$). Though this aspect has been pointed out by **McCulloch & Pitts** (1943) (as propositions) and by **Minsky & Papert** (1969) and has provided the basis for decision making with neural networks, it is important to recognize the local character of the predicate: localization influenced by the nature of the weighting function (or vector) w ($\underline{w}$) is not unrelated to the notion of diameter limitedness considered by Minsky.

Isolated neurons represent unquantified predicates of the first order logic. Continuous versions of the activation function σ initiate an

[14] Note that if x (and w) are finite dimensional, the set Ξ is isomorphic to a set containing the first n naturals, n being the number of (distinct) basis vectors necessary in describing x and w.

interpretation of neural response as predicates of first order fuzzy logic. Neurons with a response derived as monotonic activation functions operating on discriminants that are linear, due to innerproduct operation, are not capable of representing all possible predicates (of first order logic), and this limitation is overcome by networks of neurons. As these localization predicates are dependent on the relative organization of assignments in the input pattern x ($\underline{x}$), I will term the predicates arising from the influence of weighting functions as *intra-pattern predicates*.

Decisions (y) ın neural signal processors are intra-pattern predicates when attention is restricted to the influence of measurement and aggregation kernels on processor functionality. Noting that decisions of the final layer in a layered neural signal processor are dependent on those in the previous layers, the following characterization is helpful in understanding the representation of perceptually relevant operations.

$\mathcal{THEOREM}$ **6.3.2** $y^{(k)}_{\xi^{(k)}}$, $\forall \xi^{(k)} \in \Xi^{(k)}$, *in a type-k neural signal processor,* $k = 1, 2, \ . \ ,$ *is an unquantified intra-pattern predicate of hıgher order logıc (fuzzy, ıf activation functions,* σ*, are continuous) when the isolated influence of the kernels of measurement and aggregation integral transforms is considered.*

As unquantified predicates are no different from *relations*, the above statement states that neural networks are capable of representing relations, the *arity* increasing as the number of layers.

Since localization in evaluation is affected by the choice of activation functions, it is of interest to study the implications of such a localization. Noting that measurement kernels induce a foliation (possibly of non-linear manifolds, as the depth of association – *ie*, layering – increases), and the role of discrimination is to effect a reordering of the components of the foliation, neural response incorporates the result of a comparison of relative organization of assignments between members in the input pattern space. However, the activation function, in general, is not a window function, nor can always be expressed as a finite linear combination of window functions (*eg*, representation of the sigmoidal function in the linear span of window functions cannot be assured as the sigmoid is not in L^2), and, hence, *inter-pattern predicates* cannot be assured in the neural decision.

But, as established in § 6.2 (*p*. 322), directional derivatives (of all orders, assuming existence) of neural response, restricted to a linear subspace of inputs, is a superposition of window functions: for the directional derivatives to have any meaning, it is essential that the activation functions are differentiable, preferably analytic. In order to distinguish the localization due to activation functions from that due to measurement and aggregation kernels, I will term the directional derivatives of the neural response as (*directional*) *inter-pattern predicates*. While it is of interest to know the nature of inter-pattern predicates, in the sense of the system of logic incorporated, this problem has not been addressed in this thesis.

As neural signal processor responses (concepts) are derived as linear combinations of neural network decisions, it is important to note that concepts in neural signal processors refer to a superposition of labels (or signal assignments), the superposition being dictated by the predicates realized in the decision units. (This motivates the term 'aggregation kernel' for K_v.) It is immediately evident that concepts realized in neural signal processors can be taxonomical or complexive (see § 2.3) depending on the degree of overlap between the different predicates that constitute the concept. Since the concepts are derived from decisions, it is natural to expect the responses of neural signal processors to reflect localized evaluation of inter-pattern and intra-pattern relative organization of assignments.

In general, the design of a neural signal processor involves selection of activation functions as well as measurement and aggregation kernels, and, hence, a concomitant appreciation of intra-pattern and inter-pattern correlations is imperative. Noting that the space of input patterns is, in the language of category theory, a *sheaf*[15] (*cf*, **Tennison**, 1975), it is not difficult to visualize that concepts and decisions in the processing units of neural signal processors restrict evaluation to *localized regions in the sheaf of input patterns*. These regions, in general, have multiple connected components, the connectedness of each component arising from continuity in the activation functions.

[15] As the notion of a sheaf is rather involved, I will not attempt to provide a concise introduction of the same.

6.4 Kernel Influence on Representation

In an earlier section, I have shown that it is necessary for the weighting functions (weight vectors in the case of neurons on a finite-dimensional input space) to belong to a collection that is a linear combination of local functions (sequences), the requirement stemming from considerations of realization. As a consequence, the kernels of the integral transforms of measurement and aggregation are restricted to be semi-local functions. In a type-k neural signal processor, *ie*, a member of ${}^{k}\mathfrak{N}(t)$, $\forall t \in \Re_{0,+}$, the kernel $K_w^{(\ell)}(\xi^{(\ell)}, \gamma^{(\ell-1)})$ is local in $\gamma^{(\ell-1)}$, $K_\epsilon^{(\ell)}(\xi^{(\ell)}, \gamma^{(\ell)})$ is local in $\gamma^{(\ell)}$ and $K_v^{(\ell)}(\gamma^{(\ell)}, \xi^{(\ell)})$ is local in $\xi^{(\ell)}$, where $\xi^{(\ell)} \in \Xi^{(\ell)}$, $\ell = 1, 2, .. \ k$, $\gamma^{(\ell)} \in \Gamma^{(\ell)}$, $\ell = 0, 1, 2, . \ . k$, $k = 1, 2, . \ ..$

As a specific instance of kernels with a (semi) local character, I will consider the case when the kernels of the integral transforms of measurement and aggregation are chosen (or restricted) to be in the class of reproducing kernels.[16] Kernels $K(\xi, \gamma)$, wherein the indices ξ and γ belong to appropriate sets Ξ and Γ, respectively, with the property (**Aronszajn**, 1950)

$$f(\xi) = \langle K(\xi, \gamma), f(\gamma) \rangle, \tag{6.9}$$

the equality is to be understood in the L^2 sense. Note that this property requires Ξ to be the same as Γ for consistency. The collection of functions that satisfy the property indicated in Equation 6.9 (when completed to form a Hilbert space) is termed a reproducing kernel Hilbert

[16] It is simple to see that all reproducing kernels $K(\xi, \gamma)$ are local in γ from Equation 6.9 Locality of K is necessary to ensure existence of the inner product

space (RKHS) with a reproducing kernel K. In order that the RKHS is valid, $K(\xi, \gamma)$, expressed as functions of γ for every value of $\xi \in \Xi \equiv \Gamma$ should satisfy the reproducing property in Equation 6.9.

It is necessary that a bivariate function $K(\xi, \gamma)$ exhibit the following properties (*op cit*) to be a reproducing kernel of a Hilbert space consisting of real valued functions.

1. Symmetry.[17] $K(\xi, \gamma) = K(\gamma, \xi)$.
2. Non-negativity. $K(\xi, \xi) \geq 0$ and $|K(\xi, \gamma)|^2 \leq K(\xi, \xi)\, K(\gamma, \gamma)$.

A kernel satisfying the above properties has been shown (*op cit*) to be uniquely associated with a RKHS. If F denotes a Hilbert space of finite dimension, say n, containing real valued functions and $\phi_1, \phi_2, \ldots \phi_N$ are N linearly independent functions of F then the reproducing kernel of the RKHS F is given by

$$K(\xi, \gamma) = \sum_{i=1}^{N} \sum_{j=1}^{N} \beta_{ij}\, \phi_i(\xi)\, \phi_j(\gamma), \tag{6.10}$$

where $[\beta_{ij}]_{i,j=1}^{N}$ is the inverse of the Gramm matrix $[\langle \phi_i, \phi_j \rangle]_{i,j=1}^{N}$ for the system of functions $\{\phi_i\}_{i=1}^{N}$.

Consider a type-1 feed-forward neural signal processor which has the same activation function operating on all the measurements:

$$\mathfrak{y}^{(1)}_{\gamma^{(1)}}(x) = \langle K_v^{(1)}(\gamma^{(1)}, \xi^{(1)}), \sigma(\langle K_w^{(1)}(\xi^{(1)}, \gamma^{(0)}), x \rangle - \theta^{(1)}(\xi^{(1)})) \rangle, \tag{6.11}$$

[17] A reproducing kernel for a Hilbert space consisting complex valued functions is Hermitian, *ie*, $K(\xi, \gamma) = \overline{K(\gamma, \xi)}$

where x: $\Gamma^{(0)} \to \Re$ denotes the input concept (pattern) from which the (desired) newer concepts $\mathfrak{y}$ (whose domain is the same as that of the function x) are realized. The measurement kernel $K^{(1)}_w$ is chosen to be a reproducing kernel for the collection of functions $\mathfrak{X}$ – the input concept (or pattern) is drawn from this space – with the accompanying assumption (of consistency) that $\Gamma^{(0)} \equiv \Xi^{(1)}$. For convenience in analysis, I assume $\mathfrak{X}$ to be a RKHS with $K^{(1)}_w$ as the reproducing kernel.

The reproducing property of $K^{(1)}_w$ on the functions in $\mathfrak{X}$ simplifies Equation 6.11 to the following expression:

$$\mathfrak{y}^{(1)}_{\gamma^{(1)}}(x) = \langle K^{(1)}_v(\gamma^{(1)}, \xi^{(1)}), \sigma(x(\xi^{(1)}) - \theta^{(1)}(\xi^{(1)}))\rangle.$$

As a result, the collection of thresholds in the first layer, *ie*, $\theta^{(1)}(\xi^{(1)})$, $\xi^{(1)} \in \Xi^{(1)}$, is the equivalent of a template of the concept under test in the incident pattern. The comparison effected through the activation function σ is in the sense of the deviation of values in the presented concept (pattern) x from those indicated in the template $\theta^{(1)}$.[18]

Recalling the discussion in Chapters 3 and 4, the notion of reproducing kernels is analogous to that of preservance applied to the collection of weights indicated by the matrix[19] $W^{(1)} \triangleq \left[\underline{w}^{(1)}_1, \underline{w}^{(1)}_2 \, . \; . \, \underline{w}^{(1)}_{m_1}\right]^T$. This matrix is the same as the measurement kernel $K^{(1)}_w(\xi^{(1)}, \gamma^{(0)})$ with the interpretation that $\xi^{(1)}$, the index of processing nodes, takes on the

[18] In contrast, the measurement kernel K_w, in all layers, of the neural signal processors considered in Chapters 4, 5 and the earlier sections of this chapter, have the interpretation of providing a template of the patterns under test.

[19] The matrix $W^{(1)}$ has been introduced in § 5.2.

discrete values in $\{1, 2, \dots m_1\}$ and $\gamma^{(0)}$, the channel index in any processing node, is assigned values from the set $\left\{1, 2, \dots m_1 \equiv m_0 \triangleq n\right\}$. Consider the class of measurement kernels as constructed below.

$$K_w^{(1)}(\xi^{(1)}, \xi^{(1)}) \in \delta(\xi^{(1)} - \imath)\left\{2^\jmath \mid \jmath = 1, 2, \dots n\right\}, \qquad i = 1, 2, \; ..n, \tag{6.12a}$$

$$K_w^{(1)}(\xi^{(1)}, \gamma^{(0)}) \in \delta(\xi^{(1)} - i_1)\,\delta(\gamma^{(0)} - i_1)\left\{\pm 2^\jmath \mid j = 1, 2, \; ..n\right\}, \qquad i_1, i_2 = 1, 2, . \;\; n, \; \xi^{(1)} \neq \gamma^{(0)}, \tag{6.12b}$$

such that the assignments to $K_w^{(1)}$ satisfy the properties of symmetry and non-negativity indicated earlier and for every value of $\xi^{(1)}$, $\xi^{(1)} = 1, 2, \; ..n$, $|K_w^{(1)}(\xi^{(1)}, \gamma^{(0)})|$ is not the same for all values of $\gamma^{(0)}$, $\gamma^{(0)} = 1, 2, \dots n$. (See Theorem 3.1.2 (*p.* 115) for the necessity of the latter restriction on the assignments to the measurement kernel.) The following is then easily observed.

Theorem 6.4.1 *A type-*1 *neural signal processor with a measurement kernel constructed as in Expression 6.12, given n, $n = 1, 2, . \;$, is equivalent to a processor wherein the weights in the distinct nodes are the preservance weights of the discrete input space $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$, $r = 1, 2, . \, .$; $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^n$, such that for all i, $\imath = 2, 3, \dots m_1 \equiv m_0 \triangleq n$, $\|\underline{w}_\imath^{(1)}\| = \|\underline{w}_1^{(1)}\|$.*

In § 4.1 I have discussed the nature of representation in type-1 neural signal processors wherein the weights of the distinct processors belong to the class of preservance weights of the discrete space $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$. Note that a kernel defined as

$$K(\xi, \gamma) = \delta(\xi - \gamma), \qquad (6\ 13)$$

where δ indicates the dirac delta function, trivially satisfies the reproducing property indicated in Equation 6.9 (p. 344) for all functions in $L^2(\Re)$, the collection of square integrable functions defined on $\Re$ In addition, **Aronszajn** (1950) has indicated that if $K_i(\xi, \gamma)$ is the reproducing kernel of the RKHS F_i then $K(\xi, \gamma) = \sum_i K_i(\xi, \gamma)$ is the reproducing kernel of the class of functions F given by

$$F = \left\{ f = \sum_i f_i \mid f_i \in F_i \right\}.$$

These observations suggest that the kernel of measurement integral transform constructed as in Expression 6.12 is a reproducing kernel, thereby pointing to the similarity in the notions of preservance in the context of discrete spaces and that of reproducing kernels in a more general context. As an example consider the discrete kernel

$$K_w^{(1)} = \begin{bmatrix} 1 & -2 & 4 \\ -2 & 4 & 1 \\ 4 & 1 & 2 \end{bmatrix}.$$

This is a reproducing kernel and, as indicated in Table 3.1 (p. 123), each row is a preservance weight of $\mathcal{P}_r^3(\zeta, \underline{\vartheta})$, $r = 1, 2, \ldots ;$ $\zeta \in \Re_+$ and $\underline{\vartheta} \in \Re^3$.

Similarity between the notions of preservance and reproducing kernels need not be restricted to the preservance weights of $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$. As discussed in § 3.4, the notion of preservance is applicable to non-null weights in $\Re^n$ (the discrete space preserved has been termed as preservance input space). This admits a larger class of reproducing kernels, than that suggested in Expression 6.12, to be compared with a collection of preservance weights.

However, the notions of preservance in discrete spaces is not the same as the property of reproduction in the measurement kernels. The differences are due to the fact that while in preservance it is important to ensure order preservation, no such restriction is made in the case of the reproducing property of kernels. (It is simple to see that uniqueness preservation is assured in the reproduction property and the condition of regularity in preservance is introduced only for analytical convenience.) In addition, note that not all collections of preservance weights (even in the restricted case when $m_0 \triangleq n$, the number of input channels, is the same as m_1, the number of processing nodes in the type-1 neural signal processor) are reproducing kernels.

Nashed & Walter (1991) remark that to every reproducing kernel a sampling theorem is associated, *ie*, given that the kernel $K_w^{(1)}(\xi^{(1)}, \gamma^{(0)})$ is a reproducing kernel for the Hilbert space $\mathfrak{X}$ there exist (sampling) functions ${}^{w}S_\imath^{(1)}(\xi^{(1)})$, for denumerable integer values of $\imath$, such that (note that $\Xi^{(0)} \equiv \Gamma^{(1)}$ as $K_w^{(1)}$ is a reproducing kernel for $\mathfrak{X}$)

$$x(\xi^{(1)}) = \sum_\imath x_\imath \, {}^{w}S_\imath^{(1)}(\xi^{(1)}) \text{ for all } x \in \mathfrak{X}. \tag{6.14}$$

In the above equation, the equality is in the L^2 sense and all concepts (considered analogous to signals) in $\mathfrak{X}$ are assumed to be band-limited with a spectral band $[-\Omega, \Omega] \subset \Re$ and $\{x_i\}$ refers to a denumerable collection of uniformly spaced[20] samples of the incident concept x, the sampling period is decided by the Nyquist rate appropriate to a bandwidth Ω. The convolutional kernel described by Equation 6.13 (*p*. 348) is the simplest example of a sampling function

The sampling functions ${}^{w}S_i^{(1)}$ are related to the reproducing kernel $K_w^{(1)}$ in the following way (*op cit*).

$$ {}^{w}S_i^{(1)}(\xi^{(1)}) = \sum_j K_w^{(1)^{-1}}(i,j)\, K_w^{(1)}(\xi^{(1)}, j)\,, \tag{6 15}$$

where i and j take on denumerable integer values and $K_w^{(1)^{-1}}$ is the inverse of the kernel (matrix) $K_w^{(1)}(\xi^{(1)}, \gamma^{(0)})$ (restricted such that $\xi^{(1)} \in \{i\}$ $\gamma^{(0)} \in \{j\}$) as an operator on ℓ^2. (This inverse is bounded as $K_w^{(1)}$ is non-singular.) While it desired that the sampling functions form an orthonormal basis, the minimal structure required in the collection of sampling functions is biorthogonality:

$$ \langle {}^{w}S_i^{(1)}(\xi)\,, K_w^{(1)}(\xi, j) \rangle = {}^{w}S_i^{(1)}(j) = \delta(i - j)\,, \tag{6.16}$$

where i and j take on denumerable integer values. The orthogonal basis of the RKHS corresponding to the reproducing kernel $K_w^{(1)}$ is given by

$$ {}^{w}\Phi_i^{(1)}(\xi^{(1)}) = \sum_j K_w^{(1)^{-\frac{1}{2}}}(i,j)\, K_w^{(1)}(\xi^{(1)}, j)\,. \tag{6.17}$$

[20] This discussion need not be restricted to the case of uniformly sampled sequences, as the reconstruction of functions with a sequence of non-uniformly sampled values of the function is assured, under appropriate conditions, by the Paley-Wiener sampling theorem (**Benedetto** (1992))

On an incorporation of the representation of the incident concepts (patterns) x as suggested by the sampling theorem indicated in Equation 6 14, the evaluation of a type-1 neural signal processor modeled in Equation 6.11 (*p*. 345) reduces to the following expression (note the reproducing nature of the kernel $K_w^{(1)}$):

$$\mathfrak{y}_{\gamma^{(1)}}^{(1)}(\underline{x}) = \langle K_v^{(1)}(\gamma^{(1)}, \xi^{(1)}), \sigma(\sum_i x_i \, {}^w S_i^{(1)}(\xi^{(1)}) - \theta^{(1)}(\xi^{(1)})) \rangle, \qquad (6.18)$$

where $\{x_i\}$ denotes the sequence of samples of x. In the above Equation, the number of samples of the incident concept x is denumerable.

However, if in addition to band-limitedness, the concept x exhibits a locality in the domain of definition, $\Gamma^{(0)}$, *ie*, x expressed as a function of $\gamma^{(0)} \in \Gamma^{(0)}$ is in the linear span of a finite number of suitably chosen window functions, then a representation of x through a sequence of samples, as in Equation 6.14, will need no more than a finite number of (uniformly[21] spaced) samples: representation is in the sense of minimizing the $L^2(\Re)\,[\mu]$ norm (with a measure μ) of the error in approximating x with a superposition of a finite number of samples weighing the sampling functions. (See **Daubechies**, 1992, for the reasoning involved in this statement.) An implication of the conjoint spatial-spectral locality in the incident patterns, $x \in \mathfrak{X}$, follows.

[21] The adequacy of a finite number of samples for a (satisfactory) representation of signals (functions) that have a compact support in the (Fourier) spectrum and exhibit locality in the domain of definition (*eg*, time) is not limited to the case of uniform sampling. Representation of non-uniformly sampled signals is commonly addressed through the Paley-Wiener sampling theorem (**Benedetto**, 1992).

THEOREM **6.4.2** *A neural signal processor with a finite number of input elements represents an evaluation on continuous concepts when the kernel of the measurement integral transform is a reproducing kernel for the space of incident concepts*

The above statement implies that even though the inputs are restricted to a subspace $\mathcal{X}$ of the finite dimensional Euclidean space $\Re^n$, the foliations due to measurement are induced on $\mathfrak{X}$ ($\mathcal{X} \subset \mathfrak{X}$), the collection of band-limited continuous concepts exhibiting locality in the domain of definition, when the measurement kernel is a reproducing kernel for the RKHS that embeds $\mathfrak{X}$ [22] Note that the template $\theta^{(1)}$ is still assumed to have a continuous domain of definition, indicating that the collection of processing nodes is indexed on a continuous set.

Having observed the nature of representation in (type-1) neural signal processors that restrict the measurement kernel, $K_w^{(1)}$, to the class of reproducing kernels, it is natural to seek the representational characteristics of type-1 neural signal processors wherein the aggregation kernel $K_v^{(1)}$ is a reproducing kernel. For the aggregation kernel $K_v^{(1)}$ to be meaningful as a reproducing kernel, the collection of decisions $\left\{y_{\xi^{(1)}}^{(1)} \mid \xi^{(1)} \in \Xi^{(1)}\right\}$, where every member $y^{(1)}$ is expressed as a function indexed on the (continuous) set $\Xi^{(1)}$, has to be a non-trivial subset of an appropriate Hilbert space.

[22] The reproducing property of a kernel for a RKHS, say F, is inherited by every subspace of F (**Aronszajn**, 1950)

Since $y^{(1)}_{\xi^{(1)}}$, for every value of $\xi^{(1)}$ in $\Xi^{(1)}$, is derived from a function indexed in $\xi^{(1)}$ (see Equation 6.11 (*p.* 345)) through the activation function σ and as the integrability[23] of σ is not assumed in the discussion, the existence of a metric appropriate to the collection of decisions, an axiomatic requirement for a Hilbert space, cannot be assured. The influence, on representation, of reproducing property in the aggregation kernels is, thereby, not considered.

In view of the preceding discussion on the nature of representation in type-1 neural signal processors with measurement kernels of the reproducing type (see Theorem 6.4.2), it is easy to observe the following.[24] (Note that $\mathfrak{y}^0 \equiv x$, the incident concept (signal).)

THEOREM **6.4.3** *A type-k neural signal processor, $k = 2, 3, \ldots$, with discrete number of nodes in each layer represents an evaluation of continuous, band-limited and local concepts (signals) $\mathfrak{y}^{(\ell)}_{\gamma^{(\ell)}}$, $\gamma^{(\ell)} \in \Gamma^{(\ell)}$, $\ell = 0, 1, \ldots k$, when the measurement kernels $K^{(\ell)}_w$, $\ell = 1, 2, \ldots k$, are of the reproducing type.*

One of the issues that is important in a discussion involving the representation of continuous concepts (signals) through samples is that of aliasing. If the finite number of samples $\{x_i\}$ of x, in a type-1 neural signal processor, are obtained at a rate (assumed uniform, for conve-

[23] As shown in Chapter 5, sigmoidal activation functions are not integrable.

[24] The representational aspect due to the reproducing property in the measurement kernels is uninfluenced by the reproducing nature of the aggregation kernels.

nience) lesser than the minimum prescribed by the Nyquist rate for the reproducing kernel $K_w^{(1)}$, the reconstruction, $\tilde{x}$, of x through the finite number of samples will, in general, be erroneous with respect to the actual concept x. However, noting that $\tilde{x}$ is compared with the template $\theta^{(1)}$ through the activation function σ, the error, $\|\tilde{x} - x\|$, influences the decision only in a local neighbourhood of the template $\theta^{(1)}$, the extent of locality depending on the nature of the activation function σ.[25]

In the case of sigmoidal activation functions (including hard-limiter), the error due to aliasing is not appreciable when the incident concept x is at a considerable distance from θ. (This is due to the saturating nature of a sigmoidal function σ.) As a consequence, considerations of aliasing on the sampling of x (in terms of sampling rate in the case of uniform sampling) depend on the spectral characteristics of the boundary of the partition induced by the activation function θ on the RKHS embedding the input concept space $\mathfrak{X}$.

Note that the decision boundary, *ie*, the boundary of the partition, is given by the leaf of the foliation on $\mathfrak{X}$, due to the measurement kernel $K_w^{(1)}$ (assumed to be a reproducing kernel), which maps to the template $\theta^{(1)}$. But, in order to ensure that the operation of subtraction in Equation 6.18 (*p.* 351) is meaningful, the function (concept) $\theta^{(1)}$ specifying the collection of thresholds is to be a member of $\mathfrak{X}$, whence the considerations of sampling on $\mathfrak{X}$ depend on the 'concept template' $\theta^{(1)}$.

[25] See § 6 2 and § 6.3 for a discussion on the influence of σ on the local nature of processing in neural signal processors

Theorem 6.4.3 implies that considerations of sampling on the collection of concepts ${}^{\ell}\mathfrak{N}$ depend on the 'concept template' $\theta^{(\ell+1)}$, $\ell = 0, 1, \,.\, k-1$, $k = 1, 2, \,\,..$ A discussion on the nature of sampling functions, their dependence on the 'spectral' characteristics of the 'concept templates' θ and the associated issues of signal realization are not in the scope of this thesis. I now consider, briefly, the aspect of representation when the kernel $K_{\epsilon}^{(1)}$ corresponding to measurements due to lateral interaction is of reproducing type in addition to the measurement kernel $K_{w}^{(1)}$. Without any loss of generality the restricted case of type-1 neural signal processors is considered.

Arguing on the lines of feed-forward neural signal processors with measurement kernels that are of reproducing type, a type-1 neural signal processor wherein the measurement kernels $K_{w}^{(1)}$ and $K_{\epsilon}^{(1)}$ are both of reproducing type imply that the evaluation of a neural signal processor defined on the space of (continuous, band-limited and local) concepts $\mathfrak{X}$ and $\mathfrak{Y}$ (with the index sets $\Gamma^{(0)}$, $\Xi^{(1)}$ and $\Gamma^{(1)}$ being continuous) is equivalent to the following functional form.

$$\begin{aligned}\mathfrak{y}_{i^{(1)}}^{(1)}(x,\nu) \quad = \quad & \sum_{j^{(1)}} K_{v}^{(1)}(i^{(1)}, j^{(1)})\, \sigma\,(\sum_{i^{(0)}} x_{i^{(0)}}(\nu)\ {}^{w}S_{i^{(0)}}^{(1)}(j^{(1)}) \\ & + \sum_{p^{(1)}} \mathfrak{y}_{p^{(1)}}^{(1)}(x, \nu-1)\ {}^{\epsilon}S_{p^{(1)}}^{(1)}(j^{(1)}) - \theta_{j^{(1)}}^{(1)}) \qquad (6.19)\end{aligned}$$

This is identical to Equation 5.1 (p. 238)[26] with $k = 1$ when the (discrete) sampling functions ${}^{w}S_{i^{(0)}}^{(1)}(j^{(1)})$ and ${}^{\epsilon}S_{p^{(1)}}^{(1)}(j^{(1)})$, for appropriate values of $i^{(0)}$ and $p^{(1)}$, are interpreted as the feed-through and lateral interaction

[26] ν is the discrete time-travel index

connection strengths associated with the channels (indexed by $j^{(1)}$) converging on the processing nodes with indices $i^{(0)}$, $p^{(1)}$, respectively.[27] (Note that the symmetry of reproducing kernels has been invoked to arrive at this interpretation.) Thus the following is evident.

THEOREM **6.4.4** *The measurement kernels*[28] ${}^{d}K_{w}^{(\ell)}$ *and* ${}^{d}K_{\epsilon}^{(\ell)}$, $\ell = 1, 2, \ldots k$, *associated with a type-k neural signal processor,* $k = 1, 2, \ldots$, *defined with finite number of nodes represent the sampling functions corresponding to the measurement kernels, of reproducing type,* ${}^{c}K_{w}^{(\ell)}$, *and* ${}^{c}K_{\epsilon}^{(\ell)}$ *of a type-k neural signal processor that is defined to establish mappings between continuous, band-limited and local concepts (signals) through layers of continuously indexed processor arrays.*

A similar interpretation is applicable to the aggregation kernels when the activation function σ is chosen to be integrable, *eg*, Gaussian functions σ_g (see Equation 2.13 (*p.* 62)), so that the space of decisions $\{y^{(\ell)}\}$, in each layer ℓ, $\ell = 1, 2, .\ \ k$, in a type-k neural signal processor can be embedded in an appropriate Hilbert space. The above theorem suggests that the vector of connection strengths associated with every processing node is a distinct discretized (sampled) sampling function. The collection of inputs to the node denote the samples of the continuous concept (signal).

[27] This interpretation is not specific to neural signal processors with a non-null kernel of lateral interaction.

[28] The prescripts 'c' and 'd' are used to distinguish the measurement kernels corresponding to neural signal processors involving processors indexed on continuous and discrete sets, respectively.

Theorem 6.4.4 (and a similar statement regarding aggregation kernels in neural signal processors restricted to incorporate activation functions that are integrable) is a characterization of the nature of information stored (represented) in the connection strengths between the distinct processing nodes in an ensemble of neurons: this characterization is possible in the narrow, but not unrealistic, context of layered processing structures. Based on this discussion, I will now investigate the nature of representation in neural signal processors to characterize the operational aspect of neural networks.

In § 5.4, the issue of an automatic specification of weights, *ie* learning, has been interpreted, equivalently, as a design of the kernels of the integral transforms of measurement and aggregation. Such an operational scheme is based on a characterization of kernels as functions of two variables and is related to the incorporation of *a priori*, but partial, knowledge of the interconnection strengths (equivalently sampling functions) between the processing nodes. An approach to the synthesis of the kernels of the integral transforms of measurement and aggregation is indicated in Equation 6.10 (*p*. 345): the kernels are of the reproducing[29] type for a Hilbert space whose basis functions are related to the linearly independent functions $\{\phi_i\}_{i=1}^{N}$ that are involved in the realization of the kernels (see Equation 6.17 (*p*. 350).

[29] Note that if the aggregation kernels are chosen to be of the reproducing type, the activation functions should correspondingly be chosen to be of the integrable type if the internal representations are to be interpreted in terms of (non-uniform) samplings (and associated reconstruction) of the incident concepts.

Consider the measurement operation of a neural signal processor of the non-evolutionary type defined on the function space $\mathfrak{X}$.[30]

$$\eta^{(\ell)}_{\xi^{(\ell)}}(x) = \langle K^{(\ell)}_w(\xi^{(\ell)}, \gamma^{(\ell-1)}), \mathfrak{y}^{(\ell-1)}_{\gamma^{(\ell-1)}}(x)\rangle, \tag{6.20}$$

where $\xi^{(\ell)} \in \Xi^{(\ell)} \equiv \Re$, $\gamma^{(\ell-1)} \in \Gamma^{(\ell-1)} \equiv \Re$ and $\mathfrak{y}^{(0)}_{\gamma^{(0)}}(x) \triangleq x(\gamma^{(0)})$, $x \in \mathfrak{X}$. Recall from Equation 6.10 that a self-reproducing measurement kernel on a finite dimensional space is given for some ℓ, $\ell = 1, 2, \quad$, by

$$K^{(\ell)}_w(\xi^{(\ell)}, \gamma^{(\ell-1)}) = \sum_{i_1^{(\ell)}=1}^{{}_wN^{(\ell)}} \sum_{i_2^{(\ell)}=1}^{{}_wN^{(\ell)}} {}_w\beta^{(\ell)}_{i_1 i_2}\, {}_w\phi^{(\ell)}_{i_1^{(\ell)}}(\xi^{(\ell)})\, {}_w\phi^{(\ell)}_{i_2^{(\ell)}}(\gamma^{(\ell-1)}) \tag{6.21}$$

with the associated interpretation of the functions ${}_w\phi^{(\ell)}$ and the coefficients ${}_w\beta^{(\ell)}$. In the above expression, the coefficients ${}_w\beta$ and ${}_w\phi$ and the indices i have been scripted by the layer index ℓ to highlight, in the symbolization, the fact that the 'reproducing feedthrough measurement kernel basis' functions (${}_w\phi^{(\ell)}$) are not required to be the same in all the layers. The prefix 'w' is used to denote that the entities correspond to the kernel $K^{(\ell)}_w(\xi^{(\ell)}, \gamma^{(\ell-1)})$ of the integral transform of measurement due to feed-through associations. In a similar way, analogous entities corresponding to the kernels $K^{(\ell)}_\epsilon(\xi^{(\ell)}, \gamma^{(\ell-1)})$ and $K^{(\ell)}_v(\gamma^{(\ell)}, \xi^{(\ell)})$ are prefixed by 'ϵ' and 'v,' respectively.

An incorporation of the decomposition in Equation 6.21 of the measurement kernel $K^{(\ell)}_w(\xi^{(\ell)}, \gamma^{(\ell-1)})$ into the expression (Equation 6.20)

[30] Without any loss of generality, the ensuing discussion can, trivially, be extended to include an investigation on the nature of representation through the measurement kernel K_ϵ and the aggregation kernel K_v

for the measurements in layer ℓ results in the following.

$$\eta^{(\ell)}_{\xi^{(\ell)}}(x) = \sum_{i_1^{(\ell)}=1}^{{}_wN^{(\ell)}} \sum_{i_2^{(\ell)}=1}^{{}_wN^{(\ell)}} {}_w\beta^{(\ell)}_{i_1 i_2}\, {}_w\phi^{(\ell)}_{i_1^{(\ell)}}(\xi^{(\ell)})\, \langle {}_w\phi^{(\ell)}_{i_2^{(\ell)}}(\gamma^{(\ell-1)}) \,, \mathfrak{y}^{(\ell-1)}_{\gamma^{(\ell-1)}}(x) \rangle, \tag{6.22}$$

for all $\xi^{(\ell)} \in \Xi^{(\ell)}$ and $\gamma^{(\ell-1)} \in \Gamma^{(\ell-1)}$. From this equation, it is clear that the collection of measurements (η) in any layer is a representation of the concept (signal) incident on that layer. This interpretation assures that the notion of representation used in this thesis to mean a decomposition and/or synthesis of the desired functions in a 'basis' that is itself realized in accordance with the processing requirements is not invalid.[31]

In order that the above aspect is appreciated, consider the nature of measurement functions in an evolutionary neural signal processor with multiple layers. On lines similar to the derivation of Equation 6.22, the measurement functions in layer ℓ in a neural signal processor with feedthrough associations as well as lateral interactions is given by the following. (ν is the discrete time travel index.)

$$\begin{aligned} \eta^{(\ell)}_{\xi^{(\ell)}}(x,\nu) \;=\; & \sum_{i_1^{(\ell)}=1}^{{}_wN^{(\ell)}} \sum_{i_2^{(\ell)}=1}^{{}_wN^{(\ell)}} {}_w\beta^{(\ell)}_{i_1 i_2}\, {}_w\phi^{(\ell)}_{i_1^{(\ell)}}(\xi^{(\ell)})\, \langle {}_w\phi^{(\ell)}_{i_2^{(\ell)}}(\gamma^{(\ell-1)}) \,, \mathfrak{y}^{(\ell-1)}_{\gamma^{(\ell-1)}}(x,\nu) \rangle \\ & + \sum_{i_1^{(\ell)}=1}^{{}_\epsilon N^{(\ell)}} \sum_{i_2^{(\ell)}=1}^{{}_\epsilon N^{(\ell)}} {}_\epsilon\beta^{(\ell)}_{i_1 i_2}\, {}_\epsilon\phi^{(\ell)}_{i_1^{(\ell)}}(\xi^{(\ell)})\, \langle {}_\epsilon\phi^{(\ell)}_{i_2^{(\ell)}}(\gamma^{(\ell)}) \,, \mathfrak{y}^{(\ell)}_{\gamma^{(\ell)}}(x,\nu-1) \rangle, \end{aligned} \tag{6.23}$$

[31] The notion of representation is reaffirmed by the function representation theorem originating in a solution – by **Kolmogorov** (1957a) – to Hilbert's 13th problem

for all $\xi^{(\ell)} \in \Xi^{(\ell)}$, $\gamma^{(\ell-1)} \in \Gamma^{(\ell-1)}$ and $\gamma^{(\ell)} \in \Gamma^{(\ell)}$. It is of interest to note that the collection of responses (*ie*, processed concept) is a representation of neural decisions, this representation is in a basis that is chosen (or sought) for the realization aggregation kernels that are self-reproducing.

$$\mathfrak{y}^{(\ell)}_{\gamma^{(\ell)}}(x,\nu) = \sum_{i_1^{(\ell)}=1}^{{}_vN^{(\ell)}} \sum_{i_2^{(\ell)}=1}^{{}_vN^{(\ell)}} {}_v\beta^{(\ell)}_{i_1 i_2}\, {}_v\phi^{(\ell)}_{i_1^{(\ell)}}(\gamma^{(\ell)}) \left\langle {}_v\phi^{(\ell)}_{i_2^{(\ell)}}(\xi^{(\ell)}), y^{(\ell)}_{\xi^{(\ell)}}(x,\nu)\right\rangle, \quad (6.24)$$

for all $\gamma^{(\ell)} \in \Gamma^{(\ell)}$ and $\xi^{(\ell)} \in \Xi^{(\ell-1)}$.

The requirement of locality on the kernels states that measurement kernel $K^{(\ell)}_w(\xi^{(\ell)}, \gamma^{(\ell-1)})$ be a local function in the variable $\gamma^{(\ell-1)}$ for all values of the variable $\xi^{(\ell)}$. (Note that such a locality is required of the aggregation kernels also, the locality of $K^{(\ell)}_v(\gamma^{(\ell)}, \xi^{(\ell)})$ is in the variable $\xi^{(\ell)} \in \Xi^{(\ell)}$ for all values of the variable $\gamma^{(\ell)} \in \Gamma^{(\ell)}$.) In addition, the measurement kernel $K^{(\ell)}_w(\xi^{(\ell)}, \gamma^{(\ell-1)})$ is to be symmetric to be admissible as a self-reproducing kernel. These requirements stipulate that the 'reproducing feedthrough measurement kernel basis' functions ${}_w\phi^{(\ell)}$ be chosen to be local functions.

In particular, consider the 'reproducing kernel basis' functions to be the directional derivatives of a scalar valued non-evolutionary neural signal processor of type-1 with sigmoidal activation functions. Note that the class of functions realized by type-1 neural signal processors is dense in the space of continuous functions (*cf*, Theorem 5.1.1 (*p*. 243) which can be trivially extended to the case of neurons defined on function spaces): thus no generality is lost, in this preliminary investigation, by restricting attention to 'reproducing kernel basis' functions derived from functions realized by type-1 neural signal processors. The analyticity of the sigmoidal activation functions, in addition, assures that the class of functions containing the directional derivatives, in all directions $x \in \mathfrak{X}$ ($\|x\| = 1$), of the collection of functions realized by type-1 neural signal processors with sigmoidal activation functions is dense in the space of continuous functions. Proposition 6.2.4 (*p*. 330) and Theorem 6.2.1 (*p*. 332) establish that the directional derivatives, in all directions in the input space and all (positive integer) orders, of type-k, $k = 1, 2, \ldots$, neural signal processors with activation functions that satisfy the axiom of discrimination are local functions.

Let the 'reproducing feedthrough measurement kernel basis' functions be given by the following expression.

$$ {}_w\phi^{(\ell)}_{\imath^{(\ell)}}(\xi^{(\ell)}) = \frac{1}{{}_wa^{(\ell)}(\imath^{(\ell)})}(D^{\jmath}_{{}_a\tilde{x}}\tilde{\mathfrak{h}}^{(1)}(\tilde{x}))(\frac{\xi^{(\ell)} - {}_wb^{(\ell)}(\imath^{(\ell)})}{{}_wa^{(\ell)}(\imath^{(\ell)})}), \qquad (6.25) $$

for all $\xi^{(\ell)} \in \Xi^{(\ell)} \equiv \Gamma^{(\ell-1)}$, $\imath^{(\ell)} = 1, 2, .$, for some values of $\jmath$, $j = 1, 2, ..$ In the above equation, $\tilde{\mathfrak{h}}^{(1)}$ refers to a non-evolutionary type-1 neural signal processor defined on the (function) space $\tilde{\mathfrak{X}}$: the concepts (signals) drawn from this space are denoted by $\tilde{x}$ and the direction in which the differentiation is considered is denoted by ${}_a\tilde{x}$. The functions ${}_wa^{(\ell)}$ and ${}_wb^{(\ell)}$ are chosen to establish mappings between denumerable spaces, and are interpreted as the scale and translation values that the basic function $(D^{\jmath}_{{}_a\tilde{x}}\tilde{\mathfrak{h}}^{(1)}(\tilde{x}))(\xi^{(\ell)})$ is subjected to. If $\mathcal{Z}$ denotes the set of denumerable integers, then by Cantor's diagonalization argument the set $\mathcal{Z}^2 \triangleq \mathcal{Z} \times \mathcal{Z}$ is equinumerous with $\mathcal{Z}$. This aspect has encourages a singly indexed family of functions $\left\{ {}_{w}\phi^{(\ell)}_{\imath^{(\ell)}}(\xi^{(\ell)}) \right\}_{\imath^{(\ell)} \in \mathcal{Z}}$ rather than doubly indexed families of functions familiar in the discussion of wavelets. However, the necessity for distinctness between the scale and translation values has been retained through the functions ${}_{w}a^{(\ell)}$ and ${}_{w}b^{(\ell)}$.

Proposition 6.3.5 (*p*. 337) establishes that the second and higher derivatives of the sigmoidal function are basic and admissible wavelet window functions. Recall, from Proposition 6.2.4 (*p* 330), the structure of the basic function $(D^{\jmath}_{a\bar{x}}\, \tilde{\mathfrak{y}}^{(1)}(\bar{x}))(\xi^{(\ell)})$. If in this function, the constituent neural signal processor $\tilde{\mathfrak{y}}^{(1)}$ is composed as a (finite) linear combination of sigmoidal activation functions and $\jmath$ is restricted to take integer values no smaller than 2, then it is immediately evident that the 'reproducing feedthrough measurement kernel basis' functions ${}_{w}\phi^{(\ell)}_{\imath^{(\ell)}}(\xi^{(\ell)})$ are synthesized (represented) in a basis of admissible wavelets. This aspect is true of all activation functions that are continuous (analytic) and satisfy the axiom of discrimination, however, depending on the specific nature of the functional dependence, the first derivative of the activation function is also an admissible wavelet (*eg*, in the case of Gaussian activation functions, the first and higher derivatives are basic and admissible wavelet window functions.)

Note that in the representation suggested by Proposition 6.2.4 the scale factor and the translation of the wavelet window functions in the composition are governed by the dependence of the measurement kernel (actually the innerproduct of weighting functions with the (functional) direction of differentiation) and the threshold, of the non-evolutionary type-1 neural signal processor $\tilde{\mathfrak{h}}$, on the indexing variable of the decision units. The denumerable collection of admissible wavelets forms a frame when these admissible wavelets are derived, through discrete scaling and shifts in the domain, from a basic wavelet that is the second, or higher, derivative of an activation function that is continuous and satisfies the axiom of discrimination. While a proof of this statement has not been provided in this thesis, the correctness can easily be seen by recognizing the following.

Every activation function that is continuous and satisfies the axiom of discrimination has a representation in terms of a linear combination of shifted sigmoids as indicated in § 5.4. Thus the derivatives of order $\jmath$, $j = 2, 3, \ldots$, of all such activation functions are finite linear combinations of (domain) scaled and translated derivatives, of order $\jmath$, of the sigmoidal activation function. Note that the first derivative of a sigmoidal function $\sigma_s(\xi) = tanh(\xi)$ is $sech^2(\xi)$, a very good approximation to the Gaussian functions. On lines similar to that used to establish the frame property of a denumerable collection of (domain) scaled and shifted Gabor functions, a denumerable collection of functions, each given by a (domain) scaling and shift of the first derivative of the sigmoidal activation function, can be established as a frame.

Consider a denumerable collection of functions $\{\psi_i\}_{i \in \mathcal{Z}}$ and construct the denumerable collection of functions $\{\pi_i\}_{i \in \mathcal{Z}}$, where, for all $i \in \mathcal{Z}$, $\pi_i = D\psi_i$. It is then easy to establish that if all functions in the collection $\{\psi_i\}_{i \in \mathcal{Z}}$ are local and the collection is a frame, then so is the collection $\{\pi_i\}_{i \in \mathcal{Z}}$. Referring to the discussion on locality in the earlier sections of this chapter, it is evident that a denumerable collection of (domain) shifted and scaled derivatives (of second, or higher, order) of activation functions that are continuous and satisfy the axiom of discrimination satisfies the property of a frame. In addition, the functions in the collection are admissible wavelets.

Recall the representational scheme in Equation 6.22 (*p*. 359). In view of the preceding discussion on the nature of the 'reproducing feedthrough measurement kernel basis' functions $\left\{ {}_w\phi^{(\ell)}_{i^{(\ell)}}(\xi^{(\ell)}) \right\}_{i^{(\ell)} \in \mathcal{Z}}$ the measurement functions in any layer of a multi-layered neural signal processor represent the concept (signal) incident on that layer in a basis that is drawn from a wavelet frame This characterization of the nature of representation in neural signal processors is unaltered on an incorporation of the integral transform of measurement due to lateral interaction in the expression for the measurement functions $\eta^{(\ell)}$.

In a manner analogous to the measurement functions, the aggregates (responses) of any layer in a multi-layer neural signal processor is a representation, in a basis drawn from a wavelet frame, of the decisions taken on the values of the measurements functions of that layer.

In both cases, if the collection of 'reproducing kernel basis' functions forms a dual frame, then the nature of representation in neural signal processors suggests similarities with the representational nature in the conventional approach to signal processing. The measurement functions, in any layer, are a weighted average of wavelet transforms of the incident concept and the response, in any layer, of the neural signal processor is a weighted average of reconstructions, through inverse wavelet transforms, from the decisions taken on the measurement functions of the corresponding layer. This characterization of the nature of representation is specific to type-1 neural signal processors.

The above scheme can be generalized to allow the 'reproducing kernel basis' functions to be chosen to incorporate different directions of differentiation with different indices as indicated, in the case of measurement kernels, in the following expression.

$$ {}_{w}\phi^{(\ell)}_{\imath^{(\ell)}}(\xi^{(\ell)}) = \frac{1}{{}_{w}a^{(\ell)}(i^{(\ell)})}(D^{\jmath}{}_{{}_{a}\tilde{x}_{\imath^{(\ell)}}}\bar{\mathfrak{y}}^{(1)}(\tilde{x}))(\frac{\xi^{(\ell)} - {}_{w}b^{(\ell)}(\imath^{(\ell)})}{{}_{w}a^{(\ell)}(\imath^{(\ell)})}), $$

for all $\xi^{(\ell)} \in \Xi^{(\ell)} \equiv \Gamma^{(\ell-1)}$, $\imath^{(\ell)} = 1, 2, \ldots$, and for some values of $\jmath$, $\jmath = 1, 2, \ldots$. While the nature of representation is unaltered, the underlying wavelet frame in which the 'reproducing kernel basis' functions are represented incorporates rotations over the domain in addition to scaling and translation. A point of interest in the directional derivatives of neural signal processors is that they are wavelet window functions whose integrals are local (window) functions. Such a family of wavelets have been considered, in the literature, essential in a representation of signals with translation invariance.

6.5 Summary

Point-wise nonlinear associations between integral transforms characterize the representational paradigm of neural signal processors, localization in the predicates (concepts) realized by neural signal processors has been studied at two levels, one due to the nature of kernels used in the integral transforms and the other due to the nonlinear association between integral transforms. Though the kernels effective in neural signal processors between the inputs and features as well as between the decisions and concepts in multi-layered neural signal processors are, in general nonlinear, the realization of the kernels of measurement and aggregation integral transforms as cascades of nonlinearly operated kernels of linear integral transforms has allowed the study of localization due to kernels to be reduced to one of a study of localization in isolated neurons which is easily extendible to a study of localization in neural signal processors of type-1.

Localization in isolated neurons and, thereby, due to kernels of type-1 neural signal processors, studied in the case of processors defined on function (infinite dimensional vector) spaces, has been shown to be a simple consequence of the observation space (*ie*, pattern space) being an innerproduct space. In order that functions represented in isolated neurons be non-trivial, I have shown that the weight has to be in the linear span of a, possibly non-finite, collection of localized functions with vanishing asymptotes. The linear span of window functions, for every choice of window function is dense in $L^p(\Re)\,[\mu]$, $p = 1, 2, \ldots$, and any finite measure μ, as a simple consequence of Proposition 5.1.2 (*p*. 247). Dependence of features, responsible for neural response, on a local region of the incident input pattern is, thereby, assured for weights that satisfy the conventional criterion of physical realizability, that of integrability of order p, $p = 1, 2, \ldots$: in information processing contexts, square integrability, based on an equation of L^2 norms with energy, is considered adequate for realization.

Membership of the kernels of integral transforms related to measurement in a linear span of window functions has been shown to assure a localization in the support of the response of type-k neural signal processors. In addition, the kernel of aggregation integral transform that is a linear combination of window functions ensures boundedness in processor response. Localization in the support is not restricted to neurons defined on function spaces and is easily seen in the case of processors defined on finite-dimensional spaces: the analysis has been conducted mainly on infinite-dimensional spaces so that the requirement of weight belonging to the linear span of window functions is easily appreciated. Similar to the localization induced by kernels, localization is introduced in the support of neural responses by activation functions that are in the linear span of window functions: of the common choices of activation functions, Gaussian functions are window functions while sigmoidal functions, by virtue of not being in $L^2[\mu]$ for any finite measure μ, $\mu \neq 0$, cannot be represented as finite linear combinations of window functions.

Localization induced by activation functions has been studied through a characterization of directional derivatives of neural response. Activation functions satisfying the requirements indicated in the axiom of discrimination – typical examples are sigmoidal (including hard-limiter) and Gaussian functions – have been shown to exhibit the feature that all derivatives, if they exist with smoothness at the asymptotes, are window functions: this result though established only for continuous activation functions is easily extendible to discontinuous functions through the theory of generalized functions (**Hoskins**, 1979) – this has not been considered in the scope of this thesis. This property of activation functions, together with linearity of innerproducts, assures that the directional derivative of neural response is a expressed as a linear combination of window functions modulated by another localized function and, hence, localization is induced in the support of the directional derivatives.

Predicates (functions) represented by neural signal processors are local in the sense of being appropriate linear combinations of window functions and the extent of localization in the support of neural response has been identified through an analysis common in studies of function representation through window transforms. The derivatives of activation functions that are continuous and piece-wise monotonic are shown to be window functions with at least the zeroth moment vanishing, the basic requirements of wavelet windows: the first derivative might not be a wavelet window for all functions, *eg*, the first derivative of a sigmoidal function is a window function but not a wavelet window Function representation in neural signal processing, in the light of the above analysis, compares with that in conventional signal processing, however, in neural signal processing, it is not impossible to express signals and their processors in a common framework, typically that provided by the theory of wavelet transforms: however, representation in neural signal processors through wavelet transforms is not considered within the scope of this thesis.

Concepts represented in neural signal processors inherit the localized nature of the constituent predicates. As the localization in predicates are traced to kernels (of measurement and aggregation integral transforms) and activation functions, isolated consideration of the sources of localization show that the predicates represented in neural signal processors evaluate intra-pattern and inter-pattern features, the latter, in general, exhibits a directional dependence. The concepts realized in neural signal processors are derived from decisions taken on features of input space members and localization in the predicates has been shown to restrict the concept represented by each processing node to localized regions in the sheaf of input patterns.

The choice of kernels in neural signal processors influence the nature of representation and the nature of information stored in the interconnection strengths. A discussion based on kernels that are of the reproducing type shows that the interconnection strengths in neural signal processors are related to the sampling functions associated with the Hilbert space for which the kernel is reproducing. This nature of information storage suggests that conventional neural networks are capable of representing continuous concepts (signals) and processors defined over such input spaces. The nature of representation in neural signal processors has been shown to involve a function realization in a basis which is itself synthesized in a wavelet frame. Measurement functions are shown to be weighted averages of representations of the incident concept in a basis (of linearly independent functions) drawn from a wavelet frame. Similarly, the responses of neural signal processors are shown to be weighted averages of reconstructions, through inverse wavelet transforms, from decisions taken on the values of the measurement functions.

Parallel distributed processing, in the light of the above characterization of concepts, is interpreted to mean the following Distribution of representation, based on the restriction of the functionality of each processing node to local regions of the input space and a restriction of concept (function) synthesis to a local collection of decisions, is the synthesis of concepts, interpreted as neural signal processors responses, as localized evaluations of decisions on features, each of which is a measurement on local regions of the space of input patterns; localization is aimed at a discovery of features specific to the incident patterns and features common to a collection of patterns. Parallelism in representation is the requirement of simultaneous evaluation of predicates over different local regions of the ınput space, the simultaneity is necessitated more by the need to test the competing hypotheses engendered by localization of representation than the urge to gain an advantage in the complexity of computation.

Chapter 7

Representation in Neural Signal Processors: Concluding remarks

We are agreed that with our puny intelligence and understanding we can only venture so far in the great mysteries that confront us on all sides in trying to account for everything in existence and experience.

— Karl Raimund Popper and John Carew Eccles
in *The Self and Its Brain*,
Springer International, Berlin, 1977.

In this thesis, the chief concern has been a study of certain issues involved in the representation of signal processors in neural networks The focus has been one of representing signal processors abstracted to have a meaning of functional associations on the space of signals in the generic framework of neural networks, however, without imposing any specific meaning to the nature of association or the class of signals. Representation of information (signal) processors has been studied, equivalently, as function representation: different aspects of representing functions have been considered–the functions for which a representation is sought through neural processing ensembles are assumed to be defined on a multi-dimensional space and to be assigned values in a, possibly different, multi-dimensional space.

Philosophical issues involved in the connectionist approach to information processing and a review of the historical and thematic aspects of the methodological issues in the representation of signal processors through neural networks with the aim of realizing perceptually relevant information processing are presented in Chapters 1, 2 and Appendix A. The relevance of connectionist information processing in the context of an automated material handling, in particular the automated handling of information, and the scope of activity under the banner of artificial neural networks have been elaborated in Chapter 1. An attempt has been made to develop the issues relevant in a study of neural signal processing in Chapter 2.

Representation of processors in isolated neurons have been investigated in Chapter 3; this chapter has focused on processors defined on discrete spaces. The existence of weights that preserve all points of certain discrete spaces has been shown and, conversely, the existence of discrete preservance input spaces corresponding to every non-null weight has been established. Preservance has also been shown to be independent of the radix of numbering and is invariant to scaling and translation of the discrete spaces. Function representation on these discrete spaces has been reduced to sequence realization and this aspect has been incorporated in reducing learning to a procedure involving an enumeration of weights and a search, in a linearly ordered space, for the threshold. Generalization, in the sense of function extension, has been shown to influence learning by altering the function representation and, thereby, the possibility of enumerating admissible weights.

Processor realization through layered neural processing schemes has been investigated in Chapter 4: this chapter continues to focus on processors defined on discrete preservance input spaces. The adequacy of single layered neural signal processors for realizing all functions of interest has been discussed and a suggestion for architectures minimal to a given processor realization situation has been made. Learning has been shown to involve an identification of an appropriate discrete input space given a training set, analytical assignment of admissible weights, a search for threshold in a linearly ordered space and an analytical assignment for the coefficients of linear combination of neural decisions. Representation in multi-layered neural signal processors has been shown to be of the same nature as in single-layer processors, though with the possibility of a deployment of fewer neurons. An investigation into the realization of mappings between symbol spaces through neural signal processors has facilitated an algebraic characterization of the notion of linear separability: *a dichotomy over a symbol space, itself embedded in a (semi) lattice, is linearly separable if each component in the partition induced by the dichotomy on the symbol space is a sub semi-lattice.*

The representation of abstract processors with a view to understand the representational paradigm of neural signal processors has been studied in Chapter 5. Four axioms have been suggested for neural signal processing to aid a better understanding of the mechanism of representation. These axioms are sufficiently general to aid a unified study of neural signal processing architectures.

1. Axiom of Organization. A neural signal processor is composed of (layers of) three operational stages: measurement, discrimination and aggregation in that order. Preprocessing, if any, (preceding, or incorporated in, the measurement) is sought to be represented in a neural basis. Measurements are effected on an observation space constructed as the Cartesian product of the input space and a relevant subspace of a union of the space of responses of the distinct layers.

2. Axiom of Measurement. A neural signal processor, through the measurement functions in each of the processing (decision making) nodes, induces a foliation, of codimension at least one, in the input manifold. This foliation forms the basis of synthesizing (approximating) the desired level curves of the function.

3. Axiom of Discrimination A neural signal processor, through its discriminatory functions, renews the foliations, induced on the input space by the measurement functions, through a transformation, of the stems of the foliations, with at least one of the following properties:

   (a) alter the indexing of leaves to retain distinctness in a finite non-zero number of local regions of the input space,

   (b) introduce multiple components in the leaves,

   (c) associate, to at least one component of a leaf of the foliation due to discrimination, uncountably many leaves of the foliation due to measurement.

   Re-foliations provide the basis for establishing equivalences between members (elements) of the input space in ways not possible through the chosen measurement functions.

4. Axiom of Aggregation. A neural signal processor, through its aggregation function, synthesizes (or approximates) the level regions of processor response through a foliation on the Cartesian product of the stems of foliations on the input space due to discrimination.

> Concepts, in neural signal processors, are identified with the level regions of processor response.

Point-wise associations, generally nonlinear, between integral transforms has been suggested as the representational paradigm of neural signal processors. This interpretation allows a unification of neural signal processing with conventional signal processing: it is not ıncorrect to suggest that these approaches are complementary as neural signal processing is based on a search for kernels, the mechanism of association between integral transforms being invariant while conventional signal processors effect a realization through a search for an appropriate association mechanism, the nature of integral transforms being independent of the processor family. The kernels of the integral transforms of measurement and aggregation have been related to the class of kernels for nonlinear Urysohn operators and a few representational features of architectures incorporating the axioms of neural signal processing have been investigated. A study of the representation of activation functions that are continuous and satisfy the axiom of discrimination has shown that superpositions of functions with a permutation of weights are related to the issue of representation in architectures involving non-sigmoidal activation functions. This study provides an insight into the nature of representation in an ensemble of neurons wherein the weights in the nodes of a common layer are related to each other through permutation operations.

Localization characteristics in function representation through neural signal processors has been investigated in Chapter 6. Physical requirements of function realization have shown that the kernels of measurement and aggregation integral transforms are to be in the linear span of window functions. The directional derivatives of the neural signal processor response have been shown to belong to the linear span of (suitably chosen) window functions A characterization of the spectral localization has shown that the kernels of measurement and aggregation as well as the directional derivatives of neural signal processor response have a wavelet representation, thereby, allowing the possibility of a common framework for representation of signals and systems, an invaluable feature in the context of formulating Universal Neural Networks. Localization in the functionality of neural signal processors imply that the decisions evaluated by neural networks jointly exhibit the characteristics of intra-pattern predicates and inter-pattern predicates, the latter is, however, of a directional nature. Concepts in neural signal processors have been shown to represent evaluations over a localized region in the 'sheaf of input patterns.'

The characterization, in Chapter 5, of representation in neural networks as (nonlinear) point wise associations between integral transforms points to the important issue of the influence of kernel characteristics, especially localization, on representation. As a specific example, I have considered the class of neural networks wherein the kernels of the integral transforms of measurement and/or aggregation are of the reproducing type. These kernels extend the notion of preservance, applicable to processors defined over discrete spaces, to functions that are defined over continuous input spaces as well However, the notion of preservance is not the same as representation under kernels of the reproducing type: the distinction stems from the requirement of symmetry in reproducing kernels.

Representation in neural signal processors with kernels that are of the reproducing type has been shown to be equivalent to a processing situation wherein the measurements are reconstructions, of an incident concept (signal), through finitely many samples of the concept. Finitely many samples are adequate when the incident concepts belong to the class of localized signals and the samples are not restricted to be uniformly spaced. The weight vectors (weighting functions) of distinct neurons have been shown to be the distinct sampling functions associated with the Hilbert space for which the kernel, composed of the weight vectors (weighting functions), is a reproducing kernel. While the measurement kernels are easily admissible as reproducing kernels, the kernels of the integral transforms of aggregation are allowed to be of the reproducing type only if the activation functions associated with the aggregation integral transforms are integrable.

An implication of a characterization of neural network operation in terms of point-wise (nonlinear) associations between integral transforms is that the issue of learning (and generalization) is equated to a design, or selection, of the kernels of the integral transforms of measurement and aggregation. The uniqueness of the reproducing kernel for a given Hilbert space and the decomposition of reproducing kernels in a basis of linearly independent vectors (functions) allows the nature of representation in neural signal processors to be precisely established: as a consequence, the measurements in any layer are representations of the incident concepts and, in a similar way, the responses (aggregates of decisions) of a neural signal processor are representations of the decisions taken on measurements. Formally, the representational nature, in neural signal processors, has been shown to be in the sense of a weighted average of wavelet transforms

The study of localization in neural signal processors has shown that as the depth of layering increases, so does the degree of localization, *ie*, the effective receptive fields shrink with the depth of layering. While this statement has been established, largely, for activation functions that are of the sigmoidal type, the denseness of finite linear combinations of (domain) shifted and (domain) scaled sigmoids assures that this property is true of neural signal processors incorporating activation functions that are continuous and satisfy the axiom of discrimination. The nature of representation explored in this thesis allows the following characterization of the nature of representation in neural signal processors to be conjectured: 'shallow' networks are well suited for representing processors that have formal descriptions whereas 'deep' networks are necessary when the entities operating in a formal system needs to be identified/discovered Based on the characterization in Chapter 6, it would not be incorrect to suggest that symbolization (or 'symbol synthesis') involves a process of identifying, or discovering, local regions in the sheaf of input patterns, and the means of recognizing, either in isolation or in conjunction with other symbols (local regions in the sheaf of patterns), and establishing associations between symbols: the latter requirements imply a recursive usage of the object and meta-language level constructs of symbols.

In the present investigations of neural networks, the architectures are predominantly of the 'shallow' kind, *ie*, a relatively few layers are deployed, each with a large number of massively interconnected processor ensembles.[1] Architectures that are 'deep' are made of a large number of layers, each with sparsely connected processor ensembles. Such network structures will aid in implementing, in a neural basis, the requisite preprocessing (symbolization) of available signals. In terms of the axiom of measurement, symbol processors are characterized by foliations whose leaves have a relatively lesser curvature in comparison with the leaves of the foliations associated with processors that are involved in 'symbol synthesis.' As expected, the representation of a process of 'symbol synthesis' is intractable compared to the representation of 'symbol manipulation' processes. Present neuro-anatomical evidence does not seem to refute the above conjecture. The cortex and neo-cortex,

[1] Typical examples are the network structures of Hopfield, Kohonen, Grossberg, *etc*, which have a completely connected graph.

the seat of (conscious) symbolic activity, is organized as 'shallow' processors. In contrast, the mid-brain, the relatively less explored region of mental activity, is composed of 'deep' networks: this region is believed to be responsible for the (sub conscious) associations that are related to long term memory traces.

To sum up, the key findings of the attempt, in this thesis, at a characterization of representation of (signal) processors in the connectionist approach to computing are as listed in the following: no claim, however, is made as to the exhaustiveness of this study.

1. The interconnection strengths between processing nodes in an ensemble of (interconnected) neurons store knowledge of association, between inputs and outputs, by accommodating a preservation of structural regularities in the members of a certain discretely sampled subset of the input signal (pattern) space. When the input space is embedded in an Euclidean space of finite dimensions, the weights have been shown to preserve uniqueness and relative order between (input) vectors of suitably chosen discrete subsets of the input space. In contrast, the interconnection strengths between processors relate, in the case of a signal space consisting of continuous and localized signals (functions) on a continuous domain and embedded in a Hilbert space, to sampling functions associated with the reproducing kernel of the input space. The nature of information storage, in the interconnection strengths,

under the notions of preservance and the reproducing property of kernels are related, in the sense that there exists a non empty overlap between the space of kernels of the reproducing type and the collection of kernels[2] constructed through preservance weights, however, these notions are neither identical, nor is one notion reducible to the other.

2. In both forms of information storage, *ie*, the association of weights to input spaces through preservance and the selection of kernels that are reproducing in nature with respect to the input space, a knowledge of the structure in the discrete sampled subset of the input space is central to the issue of learning (and generalization), essentially a problem of kernel design. The representation of processors (functions), in the case of input spaces embedded in finite dimensional Euclidean spaces, has been shown to reduce to a process involving an identification of the preservance weight appropriate to the collection of inputs specified through the training set (*ie*, repertoire of examples), and an enumeration of the weights, in the first layer, in the class of preservance weights for the preservance input space. In contrast, processor (function) representation on signal spaces that are embedded in a Hilbert space, of continuous signals defined on a continuous domain, involves a synthesis of a collection of linearly independent 'basis' functions

[2] Kernels constructed through preservance weights are not restricted to exhibit symmetry. An imposition of the requirement of symmetry implies that preservation is effected not only in the measurement operation, but also in aggregations

through which the kernels, of a reproducing nature, are realized: these 'basis' functions, shown to belong to a wavelet frame, punctuate the character of representation in neural signal processors.

3. Operationally, the interpretation of neural signal processors effecting point-wise (nonlinear) associations between integral transforms, in the context of kernels that are chosen to be of the reproducing type, signifies that the functional character of representation in neural signal processors is one of establishing nonlinear associations between reproducing kernel Hilbert spaces. This characterization of representation in neural signal processors is of particular importance in neural networks organized to have a finite number of nodes, each operating on a finite number of inputs. In such networks, the issue of learning (and generalization) is not merely one of realizing the nonlinear association through an appropriate composition of suitably selected layers of neural processing, but also involves the crucial issue of a representation of the input signal space through adequately chosen sampling functions. The nature of representation of the input space in the measurement functions has been related, in this thesis, to the representation offered by finite linear combinations of members of a wavelet frame. I have also shown that the nature of representation in neural signal processors allows a common framework for a characterization of the representation of signals as well as processors, thereby suggesting the feasibility of an inquiry into the

theory of computation in the formalism of neural networks: in view of the distinct differences between the formalisms of Neural Networks and Turing Machines, the theory of computation in neural networks can be expected to be different from that currently established in the context of Turing Machines.[3]

4. The connectionist approach to representation of (information) processors is not restricted to the realization, or approximation, of functions on suitably defined spaces of numbers. An algebraic characterization of the principle underlying the basic processing unit, *ie*, the notion of linear separability, has been provided to show that categorization of the linearly separable kind partitions a (semi) lattice into sub semi-lattices. This characterization provides the key to designing schema of 'linear categorizers' on symbol spaces. Neural networks need not be restricted to be organized as a schema of interconnected 'linear categorizers' I have suggested four axioms that capture the essence of the prevalent architectural varieties in the connectionist approach to (information) processor representation. Of these, the axiom of measurement relates the representation of the given input signal (pattern) space with that of the nonlinear association between the input

[3] Note that in neural networks, the focus is one of seeking a representation given 'adequate' examples. It is of interest to investigate the nature of computation in neural networks *vis a vis* the currently accepted notion of computing in the framework of Turing Machines This investigation is particularly relevant in an inquiries seeking the ability of neural networks to represent decisions related to the decision making of neural networks· this inquiry needs the formulation of universal neural networks.

and output spaces. In a similar way, the axiom of aggregation (measurements are not unrelated to de aggregations) relates the representation of association between the input and output spaces with that of the desired output signal (pattern) space. The axiom of discrimination, through a consideration of the representation of (nonlinear) associations between the entities that represent the input and output signal spaces, links the isolated representations of the input and output signal spaces.

In Chapter 3, the notion of preservance has facilitated a precise characterization of the number of functions of a specified order of separability (*eg*, linear separable functions) that can be realized in any preservance weight 'direction' given the dimensionality of the input space and the index of ranking in the preservance input space. This result cannot, however, be readily used to state the exact number of functions of a specified order of separability on a preservance input space of given dimensionality and ranking even through the finite number of preservance weight 'directions' given the input space dimensionality is precisely known. This limitation ensues in view of the fact that the algebraic characteristics of the class of preservance weights are not completely known. It is, thereby, imperative that such a characterization of the class of preservance weights be investigated. This characterization will also enable a better appreciation of the issue of learning and generalization in layered neural signal processors defined on preservance input spaces.

The characterization of representation, in Chapter 4,of functions in neural signal processors defined over preservance input spaces was restricted, in processors of the multi-layered variety, to the case wherein only the weights in the first layer has the interpretation of being preservance weights of the preservance input space. While such a restriction simplifies the representation of functions on multi-variate input spaces to a situation of sequence realization on univariate spaces, the notion of preservance has not yet been fully exploited. An investigation into the nature of representation in multi-layered neural signal processors wherein the weights of the nodes in each layer is a preservance weight of collection of vectors formed by the responses of the nodes in the preceding layer is of interest to seek the computational advantages such a scheme offers in the context of learning and generalization. In addition, the inquiry in Chapter 4 has not been aimed at processors whose weights are drawn from rotated versions of the collection of preservance weights associated with some other node in the same layer: however, in Chapter 5 this problem is related, cursorily, with the issue of realizing activation functions satisfying the axiom of discrimination as finite linear combinations of shifted sigmoidal functions.

An adequate exposition outlining the nature of a theory of representation in neural signal processors does not follow the axioms of neural signal processing. The lack of such an insight, in Chapter 5, stems from the fact that while the investigation of foliations, in the context of category theory, is substantial, there does not seem to be enough characterization of foliations in terms of their stem (in fact, the set that supports an indexing of the leaves of a foliation does not seem to have been given enough consideration in the investigations). In order that the full potential of the axioms of neural signal processing be understood, it is essential to investigate foliations to provide the required characterization in terms of the stem.

Point-wise (nonlinear) associations between integral transforms, as a characterization of the operation of neural networks, allows an appreciation of the functionality of neural networks in terms of the kernels. The incorporation of available knowledge through kernels inspires newer issues, the principal one being the influence of correspondences (or correlations) in the weights (weighting functions) of distinct nodes on the representation potential. The reproducing property in the kernels, a situation that restricts the choice of weights of the incoming channels in a node to be the same as the interconnection strengths of the outgoing channels in the corresponding node of the previous layer, has been shown to state the nature of representation in terms of sampling functions. In order to aid a better understanding of the nature of representation in neural signal processors it is necessary to continue the characterization of representation in neural signal processors wherein the kernels are realized through the responses of some other, appropriately chosen, neural signal processors. This investigation will need a substantial incorporation of the representation characteristics suggested by the axioms of neural signal processing.

New Directions

In Chapter 6, the derivatives of sigmoidal activation functions, and thereby all activation functions that are continuous and satisfy the axiom of discrimination, have been shown to be localized functions satisfying the requirements of window functions. The first derivative of the sigmoidal functions is of specific interest as this functional form, the square of the hyperbolic secant function, has been extensively used in the study of nonlinear evolutionary systems, typically nonlinear wave propagation and interaction between traveling waves. In these studies, the square of the hyperbolic secant[4] is a basic solution of the popular form of the Korteweg de Vries (KdV) equation and all solutions of this equation are termed solitons (more precisely solitary waves). (See **Lamb**, 1980, **Rajaraman**, 1982, **Drazin**, 1983 and **Drazin & Johnson**, 1989 for the notion of solitary waves and solitons.)

Lax (1968) has shown that solitons are related, through a squaring operation, to the eigen functions of a Schrödinger (second order) differential operator. This aspect has been used in finding the solutions to the KdV equation through the method of *inverse scattering*.[5] The identical-

[4] The sigmoidal activation function is also a basic solution of the class of KdV equations However, this form of the KdV equation is not frequently used in the study of nonlinear evolutionary systems Evolution in systems with solitary waves of the sigmoidal kind have been considered in the investigations of von Neumann and Ulam.

[5] The KdV equation is a non-linear evolutionary equation and, hence, the solutions to the differential equation are not given by the linear span of the basic solutions. Inverse scattering relates the spatial evolution to the temporal evolution through a linear operator (this operator is associated with the Backlund transformations).

ity of the functional form of the eigen solutions of Schrodinger operators with the derivatives of the sigmoidal activation functions prompts a natural curiosity into the viability of the KdV equation governing the operational aspect of the axiom of discrimination. Of even greater interest is the applicability of the approach of inverse scattering in providing an insight into the nature of representation in neural signal processors. An isolated neuron whose activation function is a soliton has the interpretation of representing the dynamics of entities propagating in a rectilinear space-time continuum. Carrying this interpretation over to neuronal ensembles, a neural signal processor represents the dynamics of entities traveling in a curved space-time continuum, the curvature increasing as the degree of layering increases.

The investigation, in Chapter 6, into the nature of representation in neural signal processors has shown the possibility of incorporating a common framework in studies related to the representation of signals and (nonlinear) associations between signal spaces. A common framework for representing signals and their processors would be needed in studying the learning of neural signal processing through neural signal processors, *ie*, to develop the concept of *universal neural automata*. This exercise will be useful in understanding the limits of representation through the paradigm of 'learning by examples' and will enable a formulation of the notion of neural decidability. Further, a study of complex network structures, through means not entirely computational, will be of help in an identification of network structures that would be capable of an automated expression of functional characteristics that are anthropocentric and anthropomorphic: a significant step in an attempt to reach the ultimate goal of artificial intelligence.

On recognizing that neural networks belong to the larger class of processing structures involving a collection of functions indexed on lattice points, the computational basis of neural networks is seen to be identical to that used in the formalisms of Turing Machines, Finite State Machines, Grammars, Normal Algorithms, Cellular Automata, *etc.* An immediate generalization is to consider function fields over partially ordered index spaces. This abstraction raises new questions, the most important one of which relates to the interplay between inter-function interactions and macroscopic functional specificities. Such an interplay will be essential in a study of the cognitive capacity of the information processing approaches to automated intelligence. An incorporation of partial ordering in the index spaces will enable a meaningful representation of lists in neural signal processors and, thereby, facilitate studies in the understanding of automated processes with perceptual relevance.

Appendix A

Intelligence & Information Processing

All endeavors involving the 'interchange' of 'materials'—abstracted to include the isolated or combined participation of manifestations of 'matter,' 'energy' and 'information'—necessitate operations, or the operational equivalents, of decision making and/or the recognition of patterns. Collectively termed 'information processing,' these operations are sought to accentuate, as inferred (judged) by a participant (observer), the 'information content' in the signals that facilitate (and, possibly, necessitate) the 'interchange' of 'materials.'

In this Appendix I will provide a glimpse into the nature of automated intelligence and outline the two prominent traditions in the automation of intelligence. Following this I will briefly discuss the need for nonlinear methods in the processing of signals and will outline some of the interpretations that have been suggested, in the literature, to the processing of signals in the connectionist framework. These interpretations are relevant in understanding the representation potential of the neural network approach to the automation of intelligence.

A.1 Nature of Automated Intelligence

Success in the production of sustained energy which facilitated rapid industrialization, enabled a shift in investigations towards optimal means of harnessing available energy in material handling, inventory management and coordination of material flow between machines, planning and organization, in short, *operations research*. This focus has demanded extensive studies in the understanding (to aid a representation) of the nature of data, information, and knowledge, and of methods by which data should be organized to support inquiries oriented at seeking information required in the development of knowledge.

It is interesting to note that this course of events were predicted by von Neumann (*cf*, **Burks**, 1970) as evident below.

> John von Neumann pointed out that in the past, science had dealt mainly with the concepts of energy, power, force, and motion, and he predicted that "in the future science would be more concerned

with problems of control, programming, information processing, communication, organization, and systems."

The success in automating manufacturing processes with mechanization in the processing of information has triggered a new wave of activity, *ie*, the *automation of intelligence*.

Mechanized expression of intelligence is being looked at in the perspective of endowing machines with the ability of identifying objects (*eg*, tools and raw materials), and taking decisions regarding the nature and extent of processing required on identified objects. Processing of information, in automated systems, is generally through a variant of a hierarchy of (semi) autonomous interconnected processes.[1]

At one extreme lies centralized processing characterized by a single process in constant communication with all other processes operating as slaves. In contrast, the other extremity is typified by fully distributed, or *synergetic*, processing wherein no process has a total view of the system, yet a multitude of local operating criteria, incorporated in the (semi) autonomous processes, provide the necessary cohesion and cooperation to allow for stable patterns of evolution to emerge.

The interacting nature of information processing compels the participating processes to use a language, or an encoding (not necessarily a

[1] It is common to interpret processes in the restricted sense of being processors However, the abstraction is applicable to situations wherein a distinctive informative identity can be assigned to the various instances of processes. In the more sophisticated cases, processors are ascribed agental status.

formal system), for the interchange of information through states, symbols, or messages. An essential part of the processing capability is to be invested in detecting and/or estimating received information in the wake of distortions, introduced either due to insufficient precision in the response of a process (*ie*, processes with malformed messaging units), or due to interferences introduced by other coexisting information carriers (or noise sources), possibly with the capability of dominating the output of the information source of interest.

Each process participating in the processing of information should, therefore, be able to recognize the signals and messages put out by other processes (guided by the past), and to take decisions (action) in anticipation of that taken by other (competing) processes: this aspect is of importance in the design of the language, or coding scheme, used in between interacting processes. Thus, pattern recognition and (signal/ state) estimation form the essential core of information processing in automated systems, especially automation of intelligence.

Recognition of patterns, also termed (signal) detection, essentially involves an *a priori* identification of associations between *prototype* patterns and corresponding labels, or cluster (or class) memberships, and the problem at hand is to devise tests, or detectors, capable of representing the requisite class memberships:[2] available patterns are associated

[2] In this form, adaptive classification is not allowed. Further, the modulation of classification by value systems are completely ignored. Both of these are considered essential to appreciate human intellectual abilities The aspect of adaptive categorization has been considered in **Edelman** (1987) in relation to modeling of human perception.

with classes based on an appropriately chosen criterion of inter-pattern distance. Known as classification, hypothesis testing, clustering *etc*, depending on the available information regarding class memberships, and the context in which the pattern recognition task is being considered, several detection methods have been suggested, specially in the context of *statistical decision making*, wherein the signals are presumed available in a noisy context.

Pattern recognition necessarily has a finite number of classes, this requirement stems from the terminability criteria particular to algorithms (Turing Machines), discussed in the theory of computation. Issues related to the performance of pattern recognizers, apart from those of computational complexity of the testing method, dwell on the likelihood of misclassification: this issue translates to that of the error in approximating the (specified) class membership (*ie*, indicator) function.

Class membership, considered on a discrete – generally binary – scale (also called *crisp*) for long, is recently being viewed on multi-valued, even continuous, scales, and pattern recognition incorporating such measurements have been given the interpretation of capturing fuzzy rules of inference. Automation of intelligence with fuzzy rules of inference is believed to be more anthropomorphic than that achieved through crisp rules of inference.

Signal/state estimation, the other important constituent of information processing in automated systems, consists of methods to extend the notion of categorization to situations wherein the number of categories is arbitrarily large to render the approach of pattern recognition inapplicable. Estimation procedures need a knowledge of associations between regions in the signal (state) space and the space of categories, the latter could well be isomorphic to the signal (state) space on whose members the estimation procedure is being applied.

In *point* estimators, the associations sought are between signal samples, *ie*, points, and regions (subsets) of the space of categories. Estimation too, in an abstract sense, reduces to a problem of function approximation, *ie*, approximation of a function from an appropriate algebraic structure defined on the signal (state) space to another appropriately constructed algebraic structure on the space of categories.

The knowledge needed by estimators is generally supplied in terms of (parametrically expressed) likelihoods of associations, or, in the absence of evidence for deriving such information, in terms of likely relationships between (ordered) clusters in the signal space and the category of the signal (state) in relation to which clusters are sought—*ie*, non-parametric approaches In view of the nature of processing involved in estimation, these procedures have found utility in the *prediction* of a portion of a signal given some other segment of the same, typically prediction of future samples of a signal given a finite past.

In the automation of intelligence, one of the goals is to extend pattern recognition and (state) estimation to 'ideas' and 'concepts,' in addition to performing the same on objects (*ie*, signals). The information used in designing an automated system is generally termed as *knowledge base*, and the essential issue in the design of these systems lies in the representation of the knowledge base. Neural networks and classical AI differ in the way knowledge bases are 'internally' represented.

Programming, in the sense of design and implementation of algorithms, plays an important role in the representation of knowledge in classical AI, whereas in neural networks, the task of knowledge representation is studied under the metaphor of learning. It is common to find that while classical AI is directed mainly at pattern recognition, methods based on neural networks have been suggested to handle problems related to pattern classification as well as estimation.

A.2 Automation of Intelligence: Important approaches

> *Connectionist* or *PDP* models are catching on. There are conferences and new books every day, and the popular science press hails this new wave of theorizing as a breakthrough in understanding the mind (a typical example is the article in the May issue of *Sci: ence 86*, called "How we think: A new theory"). There are also, inevitably, descriptions of the emergence of Connectionism as a Kuhnian "paradigmatic" shift. (See **Schneider**, 1987, for an ex-

ample of this and for further evidence of the tendency to view Connectionism as the "new wave" of Cognitive Science)

The fan club includes the most unlikely collection of people. Connectionism gives solace both to philosophers who think that relying on the pseudo-scientific intentional or semantic notions of folk psychology (like goals and beliefs) mislead psychologists into taking the computational approach (*eg*, **Paul Churchland**, 1981; **Churchland**, 1986; **Dennett**, 1986); and to those with nearly the opposite perspective, who think that computational psychology is bankrupt because it doesn't address issues of intentionality or meaning (*eg*, **Dreyfus & Dreyfus**, 1988). On the computer science side, Connectionism appeals to theorists who think that serial machines are too weak and must be replaced by radically new parallel machines (**Fahlman & Hinton**, 1986), while on the biological side it appeals to those who believe that cognition can only be understood if we study it as neuroscience (*eg*, **Arbib**, 1975; **Sejnowski**, 1981). It is also attractive to psychologists who think that much of the mind (including the part involved in imagery) is not discrete (*eg*, **Kosslyn & Hatfield**, 1984), or who think that cognitive science has not paid enough attention to stochastic mechanisms or to "holistic" mechanisms .. and so on and on. It appeals to many young cognitive scientists who view the approach as not only anti-establishment (and therefore desirable) but also rigorous and mathematical .. . Almost everyone who is discontent with contemporary cognitive psychology and current "information processing" models of the mind has rushed to embrace "the Connectionist alternative".

> When taken as a way of modeling *cognitive archıtecture*, Connectionism really does represent an approach that is quite different from that of the Classical cognitive science that it seeks to replace . Connectionists propose to design systems that can exhibit intelligent behavior without storing, retrieving, or otherwıse operating on structured symbolic expressions The style of processing carried out in such models is thus strikingly unlike what goes on when conventional machines are computing some function
>
> The term 'Connectionist model' (like 'Turing Machine' or '[v]on Neumann machine') is thus applied to a family of mechanisms that differ in details but share a galaxy of architectural commitments

With these words[3] **Fodor & Pylyshyn** (1988a) introduce connectionism, or neural networks, in contrast to Classical AI: the other dominant approach to automated intelligence. In this section, I will trace the common history of these approaches, and briefly outline the evolution of ideas in the mechanized expression of intelligence. Despite statements to the contrary, it is nearly impossible to argue that our perception of intelligence is different from information processing, and the view that intelligence is consequent on information processıng, together with the approach of function realization, has dominated investigations in the connectionist approach to automated intelligence (see *eg*, **Rosenblatt**, 1958; **Minsky & Papert**, 1969; **McClelland**, **Rumelhart**, *et al*, 1986a).

[3] In this quotatıon, I have recoded the original citatıons to maintaın consistency with the cıtations in the rest of document

Automation of intelligence is traced uniquely to the pioneering efforts of **McCulloch & Pitts** (1943) describing the logical calculus immanent in the nervous activity While significant research had by then been accomplished as to the anatomy and even physiology of the brain, and the brain had been expressed as a composition of neurons, and the neuronal state transition studied as a function of the electro-chemical equilibration, McCulloch and Pitts were the first to recognize the logical operations (in fact operations of propositional calculus) incorporated by the very structure of neurons.

George Boole's proposal of a nice mathematical theory for an algebra of (propositional) logical operations (later named as Boolean algebra) being available, the discovery that biological neurons implement logical operations, at a time when electronics was slowly gaining ground, specially in the realization of digital computers, significant number of researchers were inspired to take up a study of biological information processing systems. It is worth mentioning that McCulloch continued his work in collaboration with Norbert Wiener in an area, which Wiener termed *Cybernetics*. Murray **Eden** (1983) traces the following.[4]

> Norbert Wiener, in his book entitled *Cybernetics, or Control and Communication in the Animal and the Machine*, did not define ex-

[4] At first glance, this quotation may seem out of context. However, it is important to recognize that the character of Artificial Intelligence, whether classical or connectionist, specially in the light of emerging trends in applications, is taking on the same aspect of cybernetics, *ie*, AI is providing a framework for choice and decision-making This aspect of AI is increasingly being incorporated into the control of mechanisms.

plicitly the word he believed he had coined .. It had, in fact, been used before· by André-Marie Ampére in his *Essai sur la philosophie des sciences* .

Ampére gave the following description of what he meant by cybernetics: "*Cybernetics* [cybernétique]. Relations between peoples, the subjects of study within .. international law and diplomacy . are only a small part of that which a good government must concern itself. Maintenance of public order, administration of laws, equitable distribution of taxes, selection of the people it must employ, and all ... other considerations ... require the continual attention of government. Choices must constantly be made, among diverse measures, about which measure is most appropriate to achieve the desired goal. Only by intensive study and comparison of the various elements that, for each choice, are provided by a knowledge of all that is relevant to the nation – its character, customs, opinions, history, religion, way of life and property, institutions, and laws – can government create the general rules of conduct that must guide it in regard to each particular case. Therefore, it is only after all the sciences that are concerned with these various factors that one must place the science in question here. I would call this science *cybernetics* from the word *κυβϵρνὴτησ*. From the restricted definition for the art of steering a vessel, cybernetics took on a meaning – even among the Greeks – of the art of steering in general ....

However, Ampére, despite his statement that he was generalizing the concept of steering, was not aware that it could be extended to the regulation of organismic behavior .. His classification of biology contained no niche for control, nor for that matter did his classification of physics.

One group to get inspired by the work of McCulloch and Pitts was that led by John von Neumann at Princeton. The structure of logical operations implemented by neurons were the source of inspiration for the basic electronic assemblies in the design of the first electronic digital computer ENIAC: these gates form the building blocks of present day computational devices too! Associating each neuron with an automata (of the finite-state kind) and recognizing the importance of interconnected ensembles of automata, von Neumann initiated the area, which he termed, *Cellular Automata* (*cf*, **von Neumann**, 1959; 1966).

Though the initial hope was to seek for an account of human intelligence in terms of such interconnected ensemble of automata, his own admissions state that the area of cellular automata is far removed from the problems crucial to capture, or explain, intelligence in mechanistic terms (*cf*, **Brink & Haden**, 1987). In passing we should recognize that the field of cellular automata, though initiated by von Neumann, owes its present existence and form to the extensive investigations of Stephen **Wolfram** (1986) and others.

Early computers though seen as manipulating numbers, strings of bits manipulated by a digital computer were soon recognized as being capable of representing anything – numbers, of course, but also features of the real world as evident in the following.

> The digital-computer field defined computers as machines that manipulated numbers. The great thing was, adherents said, that everything could be encoded into numbers, even instructions. In contrast, the scientists in AI saw computers as machines that manipulated symbols. The great thing was, they said, that everything could be encoded into symbols, even numbers. (*Cf*, **Newell, Shaw & Simon**, 1958; **Newell**, 1983; **Dreyfus & Dreyfus**, 1988.)

The particular interpretation of each neuron being an automaton, and that intelligence is biologically expressed through an interconnection of such finite-state machines, essentially symbol manipulation devices, Newell and Simon proposed their views in the famed hypothesis quoted below (*cf*, **Newell & Simon**, 1981; **Dreyfus & Dreyfus**, 1988). This hypothesis forms an essential component of classical AI.

> *Physical Symbol System Hypothesis* A physical symbol system has the necessary and sufficient means for general intelligent action.
>
> By "necessary" we mean that any system that exhibits general intelligence will prove upon analysis to be a physical symbol system. By "sufficient" we mean that any physical symbol system of sufficient size can be organized further to exhibit general intelligence.

Encouraged by this hypothesis, research effort on General problem solvers and Expert systems were initiated to *once and for all* solve the intricate problems that face all of humanity. The initial success of this automated information processing approach led John McCarthy, Marvin Minsky, Nathaniel Rochester and Claude E Shannon to formulate the program of Artificial Intelligence (*cf*, **McCorduck**, 1979).

According to this program, studies of representation of knowledge, formal methods to facilitate representation, a theory of information and its processing, *etc*, were the thrust areas of research. This formal systems approach is also known as the 'Top-Down' approach as the methodology adopted here is to satisfy, at each level of inquiry, the logically necessary tasks (operations) dictated by the goals enunciated by the previous levels – this hierarchical structure is hoped to ultimately pronounce the brain as the information structure logically necessary and sufficient for the expression of intelligence.

While significant interest had been developing in the field of classical AI, a few researchers, represented in their ideas best by Frank Rosenblatt, were interested in a very different approach. In Rosenblatt's own words (*cf*, **Rosenblatt**, 1962; **Dreyfus & Dreyfus**, 1988)

> The implicit assumption [of classical AI] is that it is relatively easy to specify the behavior that we want the system to perform, and that the challenge is then to design a device or mechanism which will effectively carry out this behavior ... [I]t is both easier and more profitable to axiomatize the *physical system* and then investigate this system analytically to determine its behavior, than

> to axiomatize the *behavıor* and then design a physical system by techniques of logical synthesis.

In this approach, also termed 'Bottom-Up' approach, no assertion is made of the brain being the only information processing structure to express intelligence: rather an appeal is being made to explore possible structures for the mechanized expression of intelligence. Rosenblatt was responsible for *Perceptrons*, machines claimed to be capable of perception in ways similar to those exhibited by human beings. The perceptrons were trained by presenting examples of the task to be learnt, and the learning (of weights) was based on a rule of learning, in biological neurons, discovered by **Hebb** (1949) which states that the interconnection strength between two (adjacent) neurons increases in proportion to the relative simultaneity of their firing patterns.[5]

[5] **Anderson & Rosenfeld** (1989), p 1–3, introducing **William James** (1989), point out that a rule similar to that proposed by Hebb, though at a macroscopıc scale, ınvolvıng neural clusters rather that just neurons, was suggested by Sir Wılliam James. This makes one ponder if such multı-scale organizational patterns exist in the braın, and such a structure can be exploited, through current knowledge about fractals (**Mandelbrot**, 1987; **Barnsley**, 1988) and scale-space filtering (**Witkin**, 1986), for a better understanding of the nature and structure of cognıtion. It would, indeed, be exciting if the search for structure–a research program aimed at a search for structure has been initiated in the field of crystal growth–is unified with the microstructure of cognition: such a unificatıon would need the language of general systems theory (**Klir**, 1977), synergetic systems (**Haken**, 1977) and self organization (**Hawkins**, 1961; **Haken**, 1983; **Kohonen**, 1984; **von Foerster & Zopf**, 1962) The extensive investigations, by physicists, relating Isıng spin systems (*ıe*, statistical thermodynamics) to neural networks (see, *eg*, **Amit**, 1989; **van Hemmen**, 1986; **van Hemmen, Grensing**, *et al*, 1988a; 1988b) suggest that the (micro) structure of cognition mıght not be unrelated to that of crystal growth, in particular, the emergence of macroscopıc (topological) symmetries.

Perceptrons have been the basis of investigations in neural networks, or connectionist networks carried out in the greater part of the latter half of this century. **McClelland**, **Rumelhart**, *et al* (1986a) describe perceptrons as in the following.

> Such machines consist of what is generally called a *retina*, an array of inputs sometimes taken to be arranged in a two-dimensional spatial layout; a set of *predicates*, a set of threshold units with fixed connections to a subset of units in the retina such that each predicate computes some local function over the subset of units to which it is connected; and one or more decision units, with modifiable connections to the predicates [6]

Rumelhart and McClelland use the term *Parallel Distributed Processing* (PDP) for the cognitive models described by neural networks (connectionist AI).

Smolensky (1990), however, differs from the idea that connectionist information processing has to conform architecturally with the brain, and in his proposal for a *Sub-Symbolic Processing Paradigm* states the following.

[6] In present terminology, Rosenblatt's perceptrons would be termed 2 layered feed forward neural network (3 layered, if layering is considered on the basis of entities in an ensemble bearing information rather than their exhibiting information processing ability).

> [T]he term "subsymbolic" is intended to suggest cognitive descriptions built up of the *constituents* of the symbols used in the symbolic paradigm [*ie*, classical AI]; these fine-grained constituents might be called *subsymbols*. Entities that are typically represented in the symbolic paradigm by symbols are typically represented in the subsymbolic paradigm by a large number of subsymbols . . Subsymbols are not operated upon by "symbolic manipulation": they participate in numerical – not symbolic computation . .
>
> Since the level of cognitive analysis adopted by the subsymbolic paradigm for formulating connectionist models is lower than the level traditionally adopted by the symbolic paradigm, for the purposes of relating these two paradigms it is often important to analyze connectionist models at a higher level; to amalgamate, so to speak, the subsymbols into symbols .. I will call the preferred level of the symbolic paradigm the *conceptual level* and that of the subsymbolic paradigm the *subconceptual level* ..
>
> The intuitive processor possesses a certain kind of connectionist architecture (which abstractly models a few of the most general features of neural networks) ..
>
> [K]nowledge in a connectionist system lies in its connection strengths .
>
> The intuitive processor is a subconceptual connectionist dynamical system that does not admit a precise formal conceptual-level description ....
>
> Given an input, a subsymbolic system outputs a set of inferences that, as a whole, gives a best fit to the input, in a statistical

> sense defined by the statistical knowledge stored in the system's connections

Rumelhart & Norman (1981) argue that

> Information [in neural networks] is not stored anywhere in particular Rather, it is stored everywhere. Information is better thought of as "evoked" than "found."

This contrast of the nature of knowledge representation in neural networks with that in classical AI is further strengthened by the following[7]

> In most models, knowledge is stored as a static copy of a pattern. Retrieval amounts to finding the pattern in long-term memory and copying it into a buffer or working memory There is no real difference between the stored representation in long-term memory and the active representation in working memory. In PDP models, though, this is not the case In these models, the patterns themselves are not stored Rather, what is stored is the *connection strengths* between units that allow these patterns to be re-created
>
> . . . .
>
> [I]f the knowledge is [in] the strengths of the connections, learning must be a matter of finding the right connection strengths so that the right patterns of activation will be produced under the right circumstances. This is an extremely important property of

[7] This quotation also facilitates an understanding of the underlying reason for the alternative labels *connectionism* and *Parallel Distributed Processing* to neural networks

> this class of models, for it opens up the possibility that an information processing mechanism could learn, as a result of tuning its connections, to capture the ınterdependencies between activations that it is exposed to the course of processing .
>
> [K]nowledge about any individual pattern is not stored in the connections of a special unit reserved for that pattern, but is distnbuted over the connections among a large number of processing units.[8] (**McClelland**, **Rumelhart**, *et al*, 1986a).

It is indeed difficult to imagine that two different, and in fact contradictory, theories, each claiming a Scientific status, and the ability to account for (human) intelligence could co-exist without controversies. A sociological history of the controversies in artificial intelligence has been studied by Mikel **Olazaran** (1993). Controversies in artificial intelligence are not new, and have not been settled.[9]

The only commonality between the two approaches to artificial intelligence, aside from their seeking for an account of similar phenomena

[8] Thıs latter statement refers to the controversy of *grandmother cells*. Readers interested ın thıs controversy could refer **Hofstadter** (1979)

[9] Indeed, a careful study of the history of the two dominant approaches to artificıal intelligence, *viz* classical AI and neural networks, would reveal the strong undercurrent of development of scıentific theories through paradigmatic revolutıons suggested by **Kuhn** (1962), **Schneider** (1987), as well as the vacillation between competıng (or contesting) scıentıfic theories proposed by George Wald (see *Scıentific Amerıcan* in 1966) The area of artificial intelligence provides a very good case for study by students of the philosophy (as well as sociology and history) of Science. It is to be noted, however, that in the limited focus that a thesis can take, such a study will not be attempted. Rather, the preceding quick comparison of the philosophical leanings of these two dominant approaches has been provıded to help an appreciation of neural networks in the proper perspective.

related to intelligence, is that both these approaches attempt at providing a physicalist, reductionist account to intelligence. However, the essential difference between the two approaches is that while classical AI seeks to maintain the Cartesian distinction between mental events and physical events, whereby the brain (or information processing structure) is merely the substrate of intelligence (**Pylyshyn**, 1984), neural networks identifies mental states with brain-states (**Churchland**, 1986), however, acknowledging the possibility of using mental states as macros for a collection, *ie* a chain, of brain-states.

Operationally, classical AI focuses on problems related to (knowledge) representation and identification of means for processing of represented knowledge, generally through rewriting rules prescribed by several forms of automata. Thus, Logic and Formal systems play a very important role in this approach to automated intelligence. Some of the prominent formalisms[10] for information processing are (Universal) Turing Machines, Generative Grammars (**Hopcroft & Ullman**, 1989), Finite State Machines (**Kohavi**, 1978), and Normal Algorithms (**Ershov & Palyutin**, 1984).

The equivalence of these different formalisms has been postulated by Church's Thesis (**Hopcroft & Ullman**, 1989; **Lewis & Papadim-**

[10] For specialized situations of information processing, as seen, *eg*, in the case of complex interconnected dynamical systems, these general formalisms are not quite convenient. Discrete Event Systems (**Wonham**, 1989) and Petri Nets (**Murata**, 1989) have been used in formalizing the information processing (typically control mechanism) in the arena of networked computing nodes

itriou, 1981). Neural networks, on the other hand, is an empirical enquiry, aimed at an identification of structures capable of representing specified information processing requirements, with the hope that a vast knowledge-base of architectures related to information processing tasks, would provide reasonable pointers to select an architecture given an information processing task.[11]

The limitations of a mechanized expression of 'intelligent behavior' commonly understood through the **Turing** (1981) test have been anticipated by **Descartes** (1960), also see **Dennett** (1988).

> It is indeed conceivable that a machine could be made so that it would utter words, and even words appropriate to the presence of physical acts or objects which cause some change in its organs; as, for example, if it was touched in some spot that it would ask what you wanted to say to it; if in another, that it would cry that it was hurt, and so on for similar things. But it could never modify its phrases to reply to the sense of whatever was said in its presence, as even the most stupid of men can do.

Technological and philosophical limitations of the two prominent approaches, *ie*, Classical AI and neural networks, to artificial intelligence

[11] This view, however, generates the apprehension and/or hope that, operationally, connectionist AI ultimately would be indistinguishable from classical AI, unless the knowledge base of connectionist architectures is sought to be represented by neural networks but, this too, in the final analysis, would need a *universal neural network* that might turn out to be no different from the existing notion of *Universal Turing Machines*

have been pointed out by several investigators. A few discussions of this important topic have been considered in the literature (see *eg*, **Partridge & Wilks** (1990)).

A.3 Nonlinear Signal Processing

Repeated attempts at incorporating the processes of automation have presented new challenges in the area of signal processing – particularly pattern recognition, decision-making and filtering (including signal estimation) – not merely in terms of speed or throughput of information processing, but also the context in which information processing is to be supported. Linear processors, with or without adaptivity, have been considered, in the not too distant past, to work, satisfactorily over a wide variety of applications, and have also been proved to be *optimal* over the class of all possible filtering operations when the signal to be processed is available in the context of an additive (white) Gaussian noise (*cf*, **Orfanidis**, 1988).

In recent times, however, linear approaches have been found unsatisfactory or inadequate in several application areas (like image and speech processing, sonar/radar signal processing, protein identification, *etc*) wherein human beings, with requisite training, have been found performing well, though not always at the desired speeds: in these applications, neither additivity nor Gaussianness of the noise can be assured. Part of the inadequacy is felt due to the fact that linear filtering, which generally involves system identification, requires an exhaustive account of the dependence of the output signal on the input signal, preferably as closed-form expressions. It may not always be possible to obtain an exhaustive account of input-output dependencies, especially when the causes for distortions in the processing are not known in sufficient operational detail.

Linear signal processing typically provides an ability to describe processors either in the domain of signal definition (commonly time and/or space) or, equivalently, in the spectral domain and in view of the unique property of integral and, thereby, spectral transforms mapping convolution in one domain to point-wise multiplication in the other, appropriate spectral transforms allow for a reduction in processing complexity. An immediate consequence is that in the linear approach to signal processing, also referred to as conventional signal processing, signals, whether analog, or digital, are described parametrically and all processing operations reduce to either an estimation of (signal) parameter(s) given [sufficiently many] observations or a manipulation of parameter(s) in the input signal space to obtain the desired kind of output signal space.

Nonlinear approaches to signal processing have been investigated in the literature, with claims of success, in an attempt to overcome limitations of the linear approach. In both linear and nonlinear approaches, processing implies an identification of a neighbourhood – a localized region – in the domain of the input signal corresponding to every point in the domain of the output signal, followed by a process of assigning, to every point in the output signal (domain), a function of assignments, to the input signal, over the corresponding neighbourhood. This dependence of the output, on localized regions of the input, facilitates a context-sensitive processing wherein the contextual information is expected to be provided by incorporation of an appropriate neighbourhood structure in the signal model.

While in linear signal processors, identification of neighbourhoods as well as evaluation of assignments to the output are linear,[12] no such imposition of linearity is made in the case of the nonlinear approaches. Historically, nonlinear signal processing has been of considerable interest since the 1980s, though these schemes were known in mathematical and statistical literature earlier. Of several approaches suggested, order statistics (typically median filtering), Volterra and Wiener series based filtering, homomorphic filters, and filtering based on morphological approaches, cellular automata or normal algorithms, are most popular: the latter two approaches are also termed *symbolic signal processing*. Except for the symbolic approaches, the others are commonly expressed in stochastic terms to tackle signals in noisy contexts.

Nonlinear signal processing is characterized by a grammatical approach, *ie*, signal spaces are identified with grammars[13] involving a

[12] Neighbourhoods are typically defined through delay or shift operators.

[13] I am stretching the notion of grammars to rule based methodologies to apply the notion to discrete symbol spaces as well as continuous spaces of numbers.

collection of production or rewriting rules – essentially formalisms specifying the mechanism of deriving assignments to a signal from relevant localized regions. Signal processing, in this approach, is considered in terms of a transformation of the grammar describing the input signal space to a desired grammar applicable to the output signal space. In passing, it is important to note that integral (spectral) transforms capture transformations between grammars, and hence, while the grammatical approach is not unique to nonlinear processors, it happens to be an (attempt at a unified) outlook in the absence of the spectral approach in nonlinear processors. The grammatical approach to signal processing has encouraged a breakaway from the tradition of parametric signal representation common in presentations of linear signal processors.

Linear approaches to signal processing have been, by far, the most frequently used. The main attractions of linear signal processors lies in the fact that linearity accommodates for ease in implementation and analysis, and it is adequate to have adders, multipliers (linear gain units) and delay units (or shifters) to implement the processor, in time (space) and/or spectral (spatial-frequency) domain. Linearity is also useful in unifying several conceptually distinct processing steps allowing for simplified, and computationally efficient processing steps. The concept of linearity having algebraic equivalents, linear signal processing has been easily extended to situations of processing involving not merely the object level signal descriptions (*ie*, data involving indexed collection of numerical assignments), but also meta level constructs like functions and functors (*cf*, **Krishnan**, 1981).

Signal processing, in the tradition of linearity, has now been extended to include signals described through sets, signals on topologies, signals on lattices, monoidal signal descriptions, and more general algebraic descriptions. The main emphasis in this approach is to concentrate on the invariances (or symmetries) preserved in the transformation, and to analyze, design and realize (synthesize) processors with the rich fund of knowledge that a search for structure would provide.

As signals are, in general, available in environments that involve perturbations, the processing by linear filters is optimal when the noise mixing with the signal is additive and has Gaussian statistics. When the mean square error criterion is used for realizing processors expected to handle signals corrupted by (additive Gaussian) noise, closed form expressions for the processors are quite easily available, for the processing model as well as the specific choice of parameters to be used with such a model.[14] (This, in itself, concisely states all the advantages of linear signal processing approaches.)

In signal processing, an important aspect to be noted is that thc approach to processor design is inevitably one of approximation (and associated representation). As in all other situations of approximation,

[14] Search based on mean squared error essentially descnbes the error evolution as a trajectory in L^2 The advantage of closed form expressions for processing and parametei selection can be expected when the evolution of error is sought in L^p, $p = 2, 3, \ldots$.

here too, we find that the ultimate criterion for acceptability (satisficability) of approximation rests in human (or biological) perception – a situation that cannot be modeled conclusively forever. In this light, it is significant to note that even as we accept a certain level of approximation, our resolution of discrimination improves with our technolgical involvement, thereby necessitating a further desire to better the present levels of approximation. This biologically grounded insatiable (and tenacious) demand for realizing the ideal has been seen in the development of signal processors too, particularly with the advent of digital processing technology (which recalls the history of automation).

A manifestation of the biological aspect of satisficability is seen in the qualification of performance of signal processors in various contexts of signal availability, particularly the statistics and operational nature of noise corruption. More specifically, the performance of linear signal processors has, of late, been considered unsatisfactory in the presence of noise which is either non-Gaussian, or is not of additive nature. Such situations are not very hard to find in the present context wherein information processing (through automation and artificial intelligence) has begun to pervade almost all aspects of our life. Examples abound in the realistic areas of computational vision, speech processing, sonar target detection *etc.*

Adaptivity in linear signal processors is one of the earliest known methods to overcome some of the shortcomings of purely linear approaches. The focus in adaptive signal processing is to consider the processors as belonging to a certain *a priori* chosen parameterized model class, and the specific parameters are adapted in accordance with the processing requirements, and as the processing is operational. In this approach, it is expected that as the processing progresses over the incident signal, sufficient enough information would be available to characterize the necessary deviation from global linearity.

In the realization of prediction filters (most popular being that due to Kalman), adaptation has been put to good use. The environment is modeled in the process of parameter adaptation and the current knowledge of the environment is used to estimate the signal values to be expected in the region(s) yet to be processed (in case of one-dimensional signal definition domain with a natural ordering, this is an estimate of a future signal sample). For the sake of completeness, it is worthwhile to remark that adaptation of parameters, formulated as a search for an optimal solution, is generally accomplished by variants of (stochastic) gradient descent – common strategies are Recursive Least Squares Approach and Least Mean Squares (also called Widrow-Hoff) Approach.[15]

Parameter adaptation in linear processors is a fruitful strategy to overcome some of the failings of purely linear signal processors. However, adaptation brings in limitations of its own, most notable being the cumulative effect, compounded by possible amplification, of errors

[15] A unified study of L^2 approaches to adaptive parameter selection, in signal processor design, has been considered by **Chaturvedi** (1994).

incurred in the processing of initial signal segments. Non-linear approaches, originally proposed by **Wiener** (1958), have been inducted into signal processing, largely since 1980s, to overcome the general limitations of linear signal processors.

The belated incorporation of nonlinear approaches in signal processing is easily traced to limitations imposed by technological issues in the realization of nonlinear operators, and analytical intractability of nonlinear systems, particularly the absence of spectral approaches to processor realization, and the inability of grouping processing stages into simpler processing modules. Order statistics (**Pitas & Venetsanopoulos**, 1990), typically median filtering homomorphic methods morphological approaches and Volterra operators (**Schetzen**, 1980) are some of the salient incorporations of nonlinearity in signal processing. Studies into the dynamics of nonlinear systems, particularly (deterministic) chaos (**Moon**, 1987), have been incorporated in the modeling of signals and also of processor characteristics (*eg*, van der Pol oscillators (*op cit*)).

Signal processing with neural networks (**Lippmann**, 1987; **Kosko**, 1992a; 1992b; **Haykin**, 1994) has been given considerable attention in the past decade, and filters realized through neural computation have been related to stack filters (an important class of nonlinear filters, *cf*, **Yin, Astola & Neuvo**, 1993b). Neural networks, for several reasons, have been considered attractive for processor realization, important among these are the approach to processor realization, and nature of processing. Given relevant examples of the required processing, as the necessary (internal) representations can be learnt, neural networks provide a framework for processor realization in situations wherein knowledge about the processing is not known in formal terms (specifically closed form expressions). The approximation provided by neural networks has been shown to be a Maximum Aposteriori Estimate (**Golden**, 1988).

A.4 Interpretations in Neural Signal Processing

Signal processing with neural networks, studied as a comparison of abstract formalisms, relies on approximating functions as a linear span of appropriately chosen basis functions, and layering, in neural networks, provides the necessary framework for design (or synthesis) of the required basis functions. In this sense, the process of learning is to address the problem of finding appropriate basis functions given the nature of the processor through examples in the training set, and, simultaneously, the extent to which these basis functions contribute to the desired function is also to be determined.

As information processing is expected in varied situations of signal availability, it is imperative that the potential of neural networks in accommodating different meanings to signals and their association be studied. In this context, I will concentrate on some of the salient interpretations that have been given to the neural information processing approach. To begin with, note that inputs, conductances, weighting values, and outputs (actions/responses) ($x_\imath$, $R_\imath^{-1}$, w_i, $i = 1, 2, \ldots n$, y, respectively), in the formal model of neurons presented in § 2.2 are allowed to be signed. Numbers corresponding to input intensities, weights, and outputs are commonly constrained to be non-negative, and a historical practice has been to partition the inputs as being excitatory or inhibitory, on the basis of the influence of isolated inputs on the output.

While this practice originates in the empirical accounts of biological expressions of information processing, it has been a recent tradition, stemming from considerations of analytical convenience, to regard values (numbers) with positive signs as corresponding to excitatory conditions and values with the opposite (*ie*, negative) sign as corresponding to inhibitory conditions: situations of inactivity are still identified with the origin (*ie*, zero) of the specific (naturally ordered) numbering system used. A specific consequence of this altered encoding is to permit inputs to switch (under conditions of learning and/or adaptation) between excitatory and inhibitory modes, possibly in contravention to expected biological principles. However, the technological significance of accommodating switched input channel modes cannot be ignored.

An inspection, of the formal model of a neuron under steady state conditions of additive dynamics, immediately reveals a framework similar to hypothesis testing (typically sign test), and in this interpretation, we also notice that the use of sigmoidal action function is equivalent to a randomization (*cf*, **Lehmann**, 1986) of a binary decision unit (*ie*, neuron with hard limiting action function). Neural networks differs from statistical hypothesis testing, at a paradigmatic level, in the way tests are constructed. The procedure of training, in neural networks, is expected to address the manner in which tests for prevailing hypotheses are constructed, though, no explicit effort (step) at identifying the hypotheses is considered necessary.

Statistical hypothesis testing, based on the 'top-down' approach characteristic of formal systems, places significant emphasis on the identification of hypotheses that would constitute a description of members in the relevant input space: the hypotheses, so identified, are used as the basis for designing necessary tests that would allow, by means of an input space characterization (model), a synthesis of the desired processor. As hypothesis testing, estimation and filtering reduce to function approximation/synthesis, neural networks provide a common/ unified framework for these important facets of signal processing.

Inter-neural interconnections, the basis of complex behavior in neuroscience, are not compelled to exhibit time-invariance in all instances of neural network models. Commonly, in neural network related literature, the interconnection strengths are identified with long term memory traces, and the neural inputs and outputs (*ie*, activations) with short term memory traces (*cf*, **Grossberg**, 1988). Time varying interconnection strengths are of interest in studies involving adaptation of interconnection strengths (as, *eg*, in Adaptive Resonance Theory of **Carpenter & Grossberg**, 1987a), and investigations, by computational bio-physicists, aimed at developing theories capable of a plausible account of learning in biological systems.

A certain degree of reluctance is exhibited, in the present trend of research in artificial neural networks, in accepting the interconnection strengths on the same footing as *programs* of digital computers. However, as the engineering relevance of neural networks is increasingly being appreciated, it would not be long before this status is indeed imputed to interconnection strengths, and rather than the present attempts to realize functions on a global time-scale, time-slicing (*ie*, time-sharing), familiar in the (nearly) simultaneous usage of digital computers by several users, would dominate the mode of function realization. Each time-slice is associated with an appropriate set of interconnection strengths, thereby allowing for efficient, time-localized, function realization and the switching of context between time-slices, based on past history, would then be an interesting problem to be tackled in the neural paradigm.[16]

Action/categorization functions play a very important role in the performance, and, consequently, the taxonomy of neural networks. Histor-

[16] This approach, while immediately advantageous in the utilization of neural networks, might be of relevance as a viable, and plausible, model of functioning in biological information processing substrate (typically brain).

ically, the hardlimiter function has been used for discrimination since the pioneering work of **McCulloch & Pitts** (1943) and is common in discourses relating the function of neural networks to formulae of propositional calculus, *ie*, functions of (crisp) Boolean logic.

Neurons with hardlimiter action functions have been studied as *threshold logic* during the 1960s (*cf*, **Cover**, 1965; **Hurst**, 1971) at a time when neural networks had not been assigned the present status of popularity/notoriety. Graded (monotonic) response, as provided by sigmoid functions, have been incorporated into the functionality of (recurrent) neural networks by **Hopfield** (1984), and have been considered essential for the automatic specification of parameters (weights) in multi-layered neural networks. (See **Rumelhart, Hinton & Williams**, 1986; **McClelland, Rumelhart**, *et al*, 1986a; 1986b; **Matheus & Hohensee**, 1987; **Hinton**, 1989; **Soucek**, 1992 for a discussion on learning in multi-layered neural networks through procedures involving a backpropagation of errors.)

As the sigmoidal action function has come to be accepted in the neural network research community, the function provided by networks of neurons incorporating sigmoidal action functions have been related to formulae of fuzzy logic. The monotonicity in discrimination provided by hardlimiter and sigmoid action functions have been held responsible for the inability of neural networks (with a single hidden layer, *eg*, perceptrons) to satisfactorily approximate desired class membership functions, and suggestions for overcoming this limitation have been made in terms of non-monotonic action functions, typically radial basis functions (*cf*, **Poggio & Girosi**, 1990).

Synaptic transmission, in real world neurons, is generally noisy, and with this consideration the rate of neural firing, depicted by the action/categorization function, has been expressed in the literature as being probabilistically related to the membrane potential, In the case of binary neurons, the probability of the neuron firing (at the higher of two frequencies, *ie*, $y = \zeta_1$) is given by

$$\begin{aligned} P(y = \zeta_1 | \eta \underline{x}, t) &= \sigma(\eta(\underline{x}, t)), \\ P(y = \zeta_0 | \eta(\underline{x}, t)) &= 1 - P(y = \zeta_1 | \eta(\underline{x}, t)), \end{aligned} \tag{A.1}$$

where, the function σ is generally of the sigmoidal type with $[\zeta_-, \zeta_+] \equiv [0, 1]$ (see *eg*, **Peretto**, 1992). In this thesis, σ denotes an abstract mapping of the membrane potential to the neural response: depending on the context, this notation will be interpreted as a deterministic decision function or a probabilistic appraisal of the (binary) neural decision.

Interpretation of neurons in stochastic terms has, however, not been limited to consideration of synaptic transmission noise. As neurons inherently support dynamics, and accommodate for excitatory and inhibitory inputs, stochastic dynamical systems, in particular, birth-death processes, are not infeasible. Independent Poisson streams of discrete pulses (inspired by models of spike trains along axons and dendritic arborescence) have been identified, in the literature, with the excitatory and inhibitory inputs, though with different (arrival) rates, and the equilibrium rate of the (non Poisson) output pulse stream (*ie*, neural activity level) are related with those of inputs.[17]

This simple demonstration relating neurons with stochastic dynamical systems suggests the feasibility of applying the paradigm of neural networks (with associated aspects of learning and generalization) to the study of networks of queues common in analysis, and design of distributed processing systems, and high-speed communication networks.[18] Pursuing this line of reasoning, it might not be infeasible to incorporate neural networks in studies of stochastic decision systems, in particular Markov decision processes (*cf*, **Derman**, 1970). Hidden Markov Models, frequently used in Classical AI for speech recognition, too relate in this sense to neural networks, and studies linking the two function synthesis approaches have been reported in the literature.

[17] In terms of the notations introduced earlier, the inputs $x_\imath$, $\imath = 1, 2, \ldots n$, are Poisson distributed pulse streams (with mutual independence); the interconnection strengths s_i, $\imath = 1, 2, \ldots n$, (and consequently, the weights $w_\imath$, $i = 1, 2, \ldots n$) control the nature of mixing of input streams to influence the membrane potential η, which plays the role of an internal state variable; the abstract amplification and translation functions a, and b, respectively control the (statistical) feedback; and the action function σ relates the equilibrium distributions of the state variable η and the output y.

[18] See **Kleinrock**, 1975; **Bertsekas & Gallager**, 1987; **Walrand**, 1988 for an analytical treatment of networks of queues, and their role in distributed processing systems and high-speed communication networks.

Appendix B

Notations

Notations relevant to all chapters

Relations

$\preceq$	A generic partial ordering relation.
$\leq$	The relation *less than or equal to.*
$\triangleq$	Equal by definition.
$\forall$	The universal quantifier.
$\exists$	The existential quantifier.

Spaces

$\emptyset$	The empty set.
$\Re$	The real number field.
$\Re^n$	Vector Space of n-tuples of real numbers.
$\Re_+$	Collection of positive reals in $\Re$, *ie*, $\Re_+ \triangleq \{x \mid x \in \Re \text{ and } x > 0\}$.

$\mathcal{X}$ Collection of inputs available for processing. It is common to find $\mathcal{X} \subseteq \Re^n$ for an appropriate value of n, $n = 0, 1, \ldots$.

$\mathcal{Y}$ Collection of labels, or values, to be output as a result of processing. The space $\mathcal{Y}$ is, in general, a compact subspace of $\Re^m$ for an appropriate value of m, $m = 0, 1,$.

$\mathfrak{X}$ Collection of input signals.

$\mathfrak{Y}$ Collection of output signals.

$\ell^p(\mathcal{A})$ The collection of sequences on the (support) set $\mathcal{A}$ that are summable in the p-th power of the absolute values. When the support set is the real number field, $\Re$, this collection is denoted by ℓ^p. (ℓ^2 is the collection of square summable sequences on $\Re$.)

$L^p(\mathcal{A})$ The collection of functions on the (support) set $\mathcal{A}$ that are integrable in the p-th power of the absolute values. When the support set is the real number field, $\Re$, this collection is denoted by L^p. (L^2 is the collection of square integrable sequences on $\Re$)

$C(\mathcal{A})$ The collection of continuous functions defined on the space $\mathcal{A}$.

$C^\infty(\mathcal{A})$ The class of analytic functions defined on the space $\mathcal{A}$.

Variables

i, j Generic indexing variables

n Dimensionality of (input) space $\mathcal{X}$, $n = 0, 1, \ldots$

$\underline{x}$ A vector denoting the collection of elements to be processed upon, $\underline{x} \in \mathcal{X}$.

$\underline{\eta}$ A vector denoting the collection of elements that represent an intermediate level of processing, $\underline{\eta} \in \Re^m$ for an appropriate value of m, $m = 0, 1, \ldots$.

$\underline{y}$ A vector denoting the collection of elements that form the outputs of the processor, $\underline{y} \in \mathcal{Y}$.

$\underline{w}$ A weight vector denoting the collection of elements that operate on corresponding channels in a neuron, $\underline{w} \in \Re^n$ where n refers to the number of channels in the neuron: the dimensionality of the collection of input patterns, $\mathcal{X}$, incident on a neuron is assumed to be the same as the number of input channels in the neuron.

$\boldsymbol{W}$ A weight matrix corresponding to a 'layer' of neurons. $\boldsymbol{W} = [\underline{w}_1, \underline{w}_2, \quad \underline{w}_m]^{\mathsf{T}}$, where $\underline{w}_\imath$, $\imath = 1, 2, . \quad m$, $m = 0, 1, . \quad$, are the weights vectors of the distinct neurons in the 'layer.' All neurons in a 'layer' are assumed to have an identical number of input channels to simplify symbolization and analysis

ζ_-, ζ_+ Limits of a connected interval denoting the acceptable values of outputs. $\zeta_-, \zeta_+ \in \Re$ such that $\zeta_- < \zeta_+$.

t, ν An (independent) variable with the connotations of tıme, $t \in [0, +\infty)$. When the time travel is restricted to discrete spaces, the notation used is ν. The variable ν is restrıcted to a space that is in one-one correspondence with the set of naturals $0, 1, \ldots$

Constants

$\underline{0}$ The zero vector. Dimension is to be read from the context.

$\underline{1}$ The vector of ones. Dimension is context dependent.

Functions

x A function denoting the signal to be processed, $x \in \mathfrak{X}$.

y A function denoting the result of processing, $y \in \mathfrak{Y}$.

μ Measure, generally Lesbegue.

$\sigma, \sigma_h, \sigma_s, \sigma_g$ Activation function. The commonly encountered types of activation functions are the hard-limiter (σ_h), sigmoidal (σ_s) and Gaussian (σ_g) functions.

Operations

$\underline{w}\,\underline{x}$ Inner-product (dot product) between vectors $\underline{w}$ and $\underline{x}$. The vectors $\underline{w}$ are assumed to belong to a common (Hilbert) space of patterns.

$\langle w, x \rangle$ Inner-product (dot product) between functions (signals) w and x. The functions w and x are assumed to belong to a common (Hilbert) space of signals

$|\ |$ Cardinality, when interpreted on sets.
Metric, when interpreted on functions.
The distinction should be clear from the context.

$\|\cdot\|$ Norm of a member (vector, function, *etc*) of an appropriate vector space When the space, say $\mathcal{X}$, is specifically indicated the norm is indicated by $\|\ \|_{\mathcal{X}}$. Where necessary, the measure, say μ, through which the norm is defined is explicitly indicated by the qualification $\|\cdot\|_{\mathcal{X}}\ [\mu]$.

$O|_S$ Restriction of an operation, O, to a region, S, smaller than the domain of O. $\mathcal{S} \subseteq \mathcal{D}_O$, where $\mathcal{D}_O$ is the domain of the operation O.

$\overline{\mathcal{A}}$ Closure of the set $\mathcal{A}$.

$\setminus$ Set difference.

$P(X = x)$ The probability of the random variable X taking on an instance x.

$\vee$ The binary operation of supremum (maximum) of the given (two) arguments.

$\wedge$ The binary operation of infimum (minimum) of the given (two) arguments.

$\circ$ The binary operation of function composition.

Notations defined in Chapter 2

Notations defined in Section 2 1[1]

$\mathit{\Xi}$ Domain of definition of the signal x. Scanning of the signal over this domain is indicated by the progression of $\xi \in \mathit{\Xi}$.

$\mathit{\Theta}$ Domain of definition of the signal y. Scanning of the signal over this domain is indicated by the progression of $\theta \in \mathit{\Theta}$.

$\mathcal{X}$ Range space of the signal x.

$\mathcal{Y}$ Range space of the signal y.

$\mathfrak{A}_x$ Algebraic structure (of appropriate kind) on $\mathcal{X}^{\mathit{\Xi}}$, the space of all signals from $\mathit{\Xi}$ to $\mathcal{X}$. ($\mathfrak{A}_x$ contains subsets of $\mathcal{X}^{\mathit{\Xi}}$.)

$\mathfrak{A}_y$ Algebraic structure (of appropriate kind) on $\mathcal{Y}^{\mathit{\Theta}}$, the space of all signals from $\mathit{\Theta}$ to $\mathcal{Y}$. ($\mathfrak{A}_y$ contains subsets of $\mathcal{Y}^{\mathit{\Theta}}$.)

[1] In this section, the symbols $\mathcal{X}$, $\mathcal{Y}$, θ, $\mathit{\Theta}$, ξ and $\mathit{\Xi}$ have an interpretation different from that in the rest of the thesis.

$\mathfrak{B}_x$ Space of measurements on signal x. (Possibly the same as $\mathfrak{A}_x$.)

$\mathfrak{B}_y$ Space of measurements on signal y. (Possibly the same as $\mathfrak{A}_y$.)

$\mathcal{N}_x(\theta)$ Neighbourhood structure in Ξ at the scan position θ, $\mathcal{N}_x(\theta) \subseteq \Xi$ for all $\theta \in \Theta$.

$\mathcal{N}_y(\theta)$ Neighbourhood structure in Θ at the scan position θ, $\mathcal{N}_y(\theta) \subseteq \Theta$ for all $\theta \in \Theta$.

$\mathfrak{a}_x(\theta)$ Assignments of x over $\mathcal{N}_x(\theta)$. (Note that $\mathfrak{a}_x(\theta)$ is a signal in $\mathcal{X}^{\mathcal{N}_x(\theta)}$ and also identifies an ordered subset (ordered by $\mathcal{N}_x(\theta)$) of $\mathcal{X}^{\Xi}$ such that $\forall\theta \in \Theta\ \mathfrak{a}_x(\theta) \in \mathfrak{A}_x$.)

$\mathfrak{a}_y(\theta)$ Assignments of y over $\mathcal{N}_y(\theta)$. (Note that $\mathfrak{a}_y(\theta)$ is a signal in $\mathcal{Y}^{\mathcal{N}_x(\theta)}$ and also identifies an ordered subset (ordered by $\mathcal{N}_y(\theta)$) of $\mathcal{Y}^{\Theta}$ such that $\forall\theta \in \Theta\ \mathfrak{a}_y(\theta) \in \mathfrak{A}_y$.)

ϕ Indexed collection of measures[2] on $\mathfrak{A}_x$, indexed by $\theta \in \Theta$.

ψ Indexed collection of measures on $\mathfrak{A}_y$, indexed by $\theta \in \Theta$.

f Mechanism (method) by which the evaluation of assignments to y are arrived at. This includes *correlations* between signals x and y.

$\mathfrak{s}$ Measure of mismatch (*ie*, (un)satisfiability).

$\mathfrak{S}$ The repertoire of distinct labels (possibly numbers) used to distinguish the possible mismatches.

[2] More precisely, ϕ and ψ are product measures on the algebraic structures $\mathfrak{A}_x$ and $\mathfrak{A}_y$ respectively. These functions measure an appropriate (desired) aspect of the relative organization of assignments in the signals x and y, *ie*, ϕ and ψ are predicates compatible to signals x and y.

y_d The desired form of signal on processing.

g An appropriate function (possibly incorporating ψ) specifying the idealized (or expected) form of processing needed.

ρ The mechanism by which comparison between the output signal and the desired or idealized signal forms is achieved In functional analytic terms, ρ is, generally, a (semi)metric, *ie*, a metric (distance function) with the axiom of unsignedness relaxed.

$\mathfrak{w}$ A window function.

$\mathfrak{b}$ A 'basic' wavelet window.

φ A scaling function.

Notations defined in Section 2 2

η Potential accumulated on the membrane of a neuron (*ie*, neuron state, also termed as post synaptic potential), $\eta \in \Re$.

$x_\imath$ Activity (input) on (dendritic) channel $\imath$, $i = 1, 2, \ldots n$, n being the number of channels, and $x_i \in \Re$.

a An abstract amplification function indicating the mechanism of modulation (decay) of the membrane potential η, $a\colon \Re \rightarrow \Re$, with the restriction that a takes non-negative values for reasons of stability.

b An abstract translation function specifying the extent of state translation in the dynamics of the membrane potential η, $b\colon \Re \rightarrow \Re$.

$s_\imath$ Interconnection strength, or synaptic efficacy, of channel $\imath$, $i = 1, 2, .\ .n$, $s_\imath \in \Re$.

y Action/response of the neuron, physiologically associated with the frequency of axonal spike generation, $y \in [\zeta_-, \zeta_+] \subset \Re$ for neurons with continuous valued outputs with appropriate values for ζ_- and ζ_+, or $y \in \{\zeta_0, \zeta_1, \ldots \zeta_c\}$, with *a priori* values $\zeta_j \in \Re$, $j = 0, 1, .. \; c$, $c = 1, 2, \ldots$ being (one less than) the number of categories, for neurons with discrete valued outputs.

σ Activation function mapping the membrane potential η to the response (axonal spike frequency) y, generally using a non-linear method (possibly with a provision for refractory time), $\sigma \colon \Re \rightarrow [\zeta_-, \zeta_+]$ for continuous valued neurons and $\sigma \colon \Re \rightarrow \{\zeta_0, \zeta_1, \ldots \zeta_c\}$ for discrete valued neurons.

C Membrane capacitance (constant amplification in conductance model), $C > 0$.

R Membrane (leakage) resistance (linear translation in conductance model), $0 < R < \infty$.

R_i^{-1} Conductance of channel i (connection strength in conductance model), $0 < |R_i^{-1}| < \infty$, $i = 1, 2, \ldots n$.

I Current applied externally (static translation in conductance model), $I \in \Re$

$\tau = RC$ Membrane charge-discharge time constant, $0 < \tau < \infty$.

$w_i = \frac{R}{R_i}$ Weighting value associated with channel i, $w_i \in \Re$, $i = 1, 2, \ldots n$

$\theta = RI$ Threshold of firing, $\theta \in \Re$.

L Number of layers in the network.

m_ℓ Number of processing nodes (*ie*, neurons) in layer ℓ, $\ell = 1, 2, .\ . L$.

$\underline{s}^{(\ell)}_{\jmath^{(\ell)}}$ Feed-through synaptic efficacies (*ie*, inter-layer interconnection strengths) for processing node $\jmath^{(\ell)}$ in layer ℓ, $\jmath^{(\ell)} = 1, 2, \ldots m_\ell$, $\ell = 1, 2, .\quad L$.

$\underline{\epsilon}^{(\ell)}_{\jmath^{(\ell)}}$ Feed-back (recurrent) synaptic efficacies (*ie*, intra-layer interconnection strengths) for processing node $\jmath^{(\ell)}$ in layer ℓ, $\jmath^{(\ell)} = 1, 2, .\ . m_\ell$, $\ell = 1, 2, \ldots L$.

$\tau^{(\ell)}_{f_{\jmath^{(\ell)}} \imath_f}$ Propagation (and refractory) delay in the feed-through path from processing node $\imath_f$ of layer $(\ell-1)$ to processing node $\jmath^{(\ell)}$ in layer ℓ, $i_f = 1, 2,\ .\ m_{\ell-1}$, $\jmath^{(\ell)} = 1, 2, \ldots m_\ell$, $\ell = 1, 2, \ldots L$

$\tau^{(\ell)}_{r_{\jmath^{(\ell)}} \imath_r}$ Propagation (and refractory) delay in the feed-back path from processing node i_r to processing node $\jmath^{(\ell)}$, both in layer ℓ, $i_r = 1, 2, \ldots m_\ell$, $j^{(\ell)} = 1, 2, \ldots m_\ell$, $\ell = 1, 2, .\ . L$.

Notation defined in Section 2 3

$P_\imath(\underline{x})$ A homogeneous polynomial of degree i in the elements of the vector $\underline{x}$.

Notations defined in Chapter 3

Notations defined in Section 3.1

$\mathcal{B}^n$ Collection of all n dimensional binary vectors (the elements are in $[-1, +1]$). This is also referred to as the Boolean space of n dimensions.

$\mathcal{B}^n(\zeta, \underline{\vartheta})$ Generalized Boolean space of n dimensions derived by scaling and translating $\mathcal{B}^n$. $\zeta \in \Re_+$ is the scale factor and $\underline{\vartheta} \in \Re^n$ denotes the extent of translation.

$\mathcal{L}_{\underline{w}}$ Linear sub-space in $\Re^n$ in the direction of $\underline{w}$. $\mathcal{L}_{\underline{w}} = \{\beta\underline{w} \mid \beta \in \Re\}$.

$\wp_n$ Collection of preservance weights corresponding to a discrete input space of n dimensions.

$\wp_n(\alpha)$ Restriction of $\wp_n$ to weights having α as the common factor of all elements.

$\underline{w}_{<\mathfrak{e}>}$ Enumerated preservance weight ($\mathfrak{e}$ is the enumeration index).

$\boldsymbol{P}_{\mathfrak{e}_2\mathfrak{e}_1}$ A permutation operation relating the transformation of a preservance weight $\underline{w}_{<\mathfrak{e}_1>}$ to the preservance weight $\underline{w}_{<\mathfrak{e}_1>}$, $\underline{w}_{<\mathfrak{e}_1>}, \underline{w}_{<\mathfrak{e}_1>} \in \wp_n(\alpha)$, for some $\alpha \in \Re_+$.

$\mathcal{S}_r^n(\zeta, \underline{\vartheta})$ A discrete subset of $\Re^n$.

$\mathcal{P}_r^n(\zeta, \underline{\vartheta})$ A discrete subset of rank r, $r = 1, 2, \ldots$, in $\Re^n$ useful in the study of input space preservation in isolated neurons. $\zeta \in \Re_+$ is the scale factor and $\underline{\vartheta} \in \Re^n$ is the translation. $\mathcal{P}_1^n(\zeta, \underline{\vartheta}) \triangleq \mathcal{B}^n(\zeta, \underline{\vartheta})$ for all $n = 1, 2,\ ..$, $\zeta \in \Re_+$ and $\vartheta \in \Re^n$.

$\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{B}^n(\zeta, \underline{\vartheta}))$ Preservation points in $\mathcal{L}_{\underline{w}}$ under a weight $\underline{w}$ corresponding to $\mathcal{B}^n(\zeta, \underline{\vartheta})$.

$\mathfrak{L}_{\underline{w}}(\alpha, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$ Preservation points in $\mathcal{L}_{\underline{w}}$ under a weight $\underline{w}$ corresponding to $\mathcal{P}_r^n(\zeta, \underline{\vartheta})$. When the norm of the preservance weights is fixed, then α is replaced by $\|\underline{w}\|$. The resulting notation is $\mathfrak{L}_{\underline{w}}(\|\underline{w}\|, \mathcal{P}_r^n(\zeta, \underline{\vartheta}))$.

Notations defined in Section 3.2

$\underline{\theta}$ Value of threshold in a comparison.

${}^{\underline{w}}\Theta^n_r(\alpha,\zeta,\underline{\vartheta})(\imath)$ The smallest (connected) interval in $\Re$ between the $\imath$ th and $i+1$ th ordered points in $\mathcal{L}_{\underline{w}}(\alpha,\mathcal{P}^n_r(\zeta,\underline{\vartheta}))$.

Notations defined in Section 3.3

$\mathcal{T}_\imath$ Collection of inputs listed in the training set. $\mathcal{T}_\imath \subseteq \mathcal{P}^n_r(\zeta,\underline{\vartheta})$. For the paradıgm of learning by examples to be meaningful, $\mathcal{T}_\imath \neq \emptyset$.

$\mathcal{L}_{\underline{w}}(\alpha,\mathcal{T}_\imath)$ Preservation points in $\mathcal{L}_{\underline{w}}$ of $\mathcal{T}_\imath$ under a preservance weight $\underline{w} \in \wp_n(\alpha)$ for any $\alpha \in \Re_+$.

$\mathfrak{D}$ Dichotomy on $\mathcal{P}^n_r(\zeta,\underline{\vartheta})$.

$\underline{w}\ \mathcal{A}$ The collection of projections of the vectors in the set $\mathcal{A}$ along the vector $\underline{w}$, ıe, $\underline{w}\cdot\mathcal{A} = \{\underline{w}\ \underline{x} \mid \underline{x} \in \mathcal{A}\}$. The vector $\underline{w}$ and the vectors in $\mathcal{A}$ belong to a common innerproduct (Hilbert) space.

Notations defined in Section 3 4

${}_{\mathfrak{r}}\mathcal{H}^n_r(\zeta,\underline{\vartheta})$ The totality of (scaled and shifted) radix $\mathfrak{r}$ vectors in $\Re^n$, $\mathfrak{r} = 2,3,\ldots\ .\ \zeta \in \Re_+$ is the scale factor and $\underline{\vartheta} \in \Re^n$ is the translation.

${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ A discrete subset of rank r, $r = 1,2,\ldots$, in $\Re^n$ useful in the study of preservation of an input space of radix $\mathfrak{r}$, $\mathfrak{r} = 2,3,\ .\ .$, vectors in isolated neurons. $\zeta \in \Re_+$ is the scale factor and $\underline{\vartheta} \in \Re^n$ is the translation.

${}^{\underline{w}}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ The discrete space ${}_{\mathfrak{r}}\mathcal{P}^n_r(\zeta,\underline{\vartheta})$ rotated in such a way that any vector in $\mathcal{L}_{\underline{w}} \setminus \{\underline{0}\}$ is a preservance weight of the discrete input space.

Notations defined in Chapter 4

Notations defined in Section 4 1

$\mathfrak{y}(\underline{x})$ The response of a single layer neural signal processor operating on an input pattern $\underline{x}$. For all $\underline{x} \in \mathcal{X} \subseteq \Re^n$, $\mathfrak{y}(\underline{x}) \in \Re$.

$\mathfrak{y}$ An array denoting the responses of a single layer neural signal processor to the input vectors in the discrete space $\frac{w}{\tau}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$.

$\boldsymbol{y}$ A matrix denoting the responses of the neurons participating in a single layer neural signal processor for inputs in the preservance input space $\frac{w}{\tau}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$.

$\boldsymbol{A}^*$ The Moore-Penrose pseudo inverse of a matrix $\boldsymbol{A}$.

Notations defined in Section 4 2

$\mathfrak{y}^{(\ell)}(\underline{x})$ The response in layer ℓ of a multi-layered neural signal processor operating on an input pattern $\underline{x}$. For all $\underline{x} \in \mathcal{X} \subseteq \Re^n$, $\mathfrak{y}^{(\ell)}(\underline{x}) \in \Re$.

$\underline{w}^{(\ell)}_{j^{(\ell)}}$ The weight vector associated with the $j^{(\ell)}$-th processing node in layer ℓ.

$\theta^{(\ell)}_{j^{(\ell)}}$ The threshold associated with the $j^{(\ell)}$-th processing node in layer ℓ.

$\mathfrak{y}^{(k)}$ An array denoting the responses in the k-th layer, $k = 1, 2, \ldots,$ of a k layer neural signal processor to the input vectors in the discrete space $\frac{w}{\tau}\mathcal{P}^n_r(\zeta, \underline{\vartheta})$.

${}^{\mathfrak{z}k}\boldsymbol{y}$ A matrix denoting the responses of the neurons in the k-th layer participating in a k layer neural signal processor

	for inputs in the preservance input space ${}^{w}_{\tau}\mathcal{P}^{n}_{r}(\zeta,\underline{\vartheta})$. The number of sign-transitions in the responses of individual neurons are restricted to be no more than $\mathfrak{z}_k$.

Notations defined in Section 4.4

$\mathfrak{P}(\mathcal{A})$	The partition induced on the set $\mathcal{A}$.
$\mathfrak{P}_\imath(\mathcal{A})$	The distinct members of the partition on the set $\mathcal{A}$, $\imath = 1, 2, \, ..|\mathfrak{P}(\mathcal{A})|$.

Notations defined in Chapter 5

Notations defined in Section 5.1

$\mathfrak{y}^{(k)}$	A type-k neural signal processor (indicated by the response).
${}^k\mathfrak{N}$	The collection of functions realized by a neural signal processor of type-k.
${}^k\mathfrak{N}(t)$	The collection of functions realized by a type-k neural signal processor, as a consequence of possible evolution, at time t, ${}^k\mathfrak{N}(t) \subseteq {}^k\mathfrak{N}$.
${}^k\mathfrak{N}_{\underline{m}^{(k)}}(t)$	The collection of evolutionary neural functions realized by type-k neural signal processors whose nodes are as specified in the vector $\underline{m}^{(k)}$.
$\mathcal{C}$	An arbitrary compact subset of $\mathfrak{R}$.
χ, π	Component functions in the function representation scheme of Kolmogorov.

Notations defined in Section 5 2

$\mathfrak{F}, {}^m\mathfrak{F}, {}^d\mathfrak{F}, {}^a\mathfrak{F}$	A foliation on the input space. The prefixes 'm,' 'd' and 'a' denote foliations due to measurement, discrimination and

aggregation operations, respectively. A foliation is essentially a partition wherein all members are indexed. Each (indexed) member of the partition is termed a leaf of the foliation.

$\mathcal{A}_{\mathfrak{F}}$ Stem of the foliation $\mathfrak{F}$. This is the index set for the leaves of the foliation $\mathfrak{F}$.

Notations defined in Section 5.3

x A function(al) denoting the pattern incident on the decision unit(s), x: $\Xi \to \mathfrak{X}$, such that $x(\xi)\cdot\ \emptyset \to \mathcal{X}$ for all $\xi \in \Xi$,

y A function(al) denoting the response pattern of the neural signal processor, y: $\Xi \to \mathfrak{Y}$, such that $y(\xi)$: $\mathcal{X} \times \mathcal{R}_{0,+} \to \mathcal{Y}$ for all $\xi \in \Xi$,

w Weight function in a neuron defined on a function space $\mathfrak{X}$. $w \in \mathfrak{W}$, a function space which is embedded in the same Hilbert space as $\mathfrak{X}$.

v Combining function in a neural signal processor defined on a function space $\mathfrak{X}$. $v \in \mathfrak{W}$, a function space which is embedded in the same Hilbert space as $\mathfrak{X}$.

$\boldsymbol{K}_w, \boldsymbol{K}_\epsilon, \boldsymbol{K}_v$ Kernels of the integral transforms of measurement and aggregation. $\boldsymbol{K}_w$ is the kernel of the integral transform due to feed-through associations in the measurement process, while $\boldsymbol{K}_\epsilon$ is the kernel of the integral transform of measurement due to lateral interaction. $\boldsymbol{K}_v$ is the kernel of the integral transform due to aggregation.

Notations defined in Chapter 6

Notations defined in Section 6 1

D The operator of ordinary differentiation with respect to the independent variable.

$\hat{f}$ The Fourier transform of a function f, $f \in L^1$.

ω A variable whose reciprocal has the connotations of recurrence (periodicity).

τ An independent variable denoting traversal in the domain of definition of the weighting functions w.

Notatıons defined in Section 6 2

$D^{\jmath}$ Ordinary differentiation operator of order j, $j = 1, 2, \ldots$

$\mathcal{L}_{x_a}$ A one-dimensional linear subspace in the 'direction' of the pattern (signal) x_a, $x_a \in \mathfrak{X}$.

$D^{\jmath}_{x_a}$ The operation of directional differentiation of order j, $\jmath = 1, 2, .\ .$, in the direction of the pattern (signal) x_a, $x_a \in \mathfrak{X}$, $\|x_a\| = 1$.

Notations defined in Section 6.3

$\mathfrak{I}_f$ Region of localization in the domain of definition of the function f.

Δ_f Width of the region of localization in the domain of definition of the function f.

Notations defined in Section 6.4

$\{\phi\}_{\imath=1}^{n}$ A collection of n, $n = 1, 2, .\ $, linearly independent functions. The kernel of a reproducing kernel Hilbert space is represented as certain linear combinations of these 'basis' functions.

δ The Dirac delta function.

Ω The bandwidth of band-limited signals (concepts).

${}^{w}S^{(\ell)}(\cdot)$ Sampling functions in layer ℓ associated with the integral transform of (feed-through) measurement.

${}_{w}\phi^{(\ell)}_{i^{(\ell)}}$ A collection of linearly independent functions that are used to realize self-reproducing kernels of the integral transforms of measurement due to feedthrough associations in the ℓ-th layer of a multi-layer neural signal processor. $i^{(\ell)} = 1, 2, . . {}_{w}N^{(\ell)}$, ${}_{w}N^{(\ell)} = 1, 2, \ldots$. These functions are termed 'reproducing feedthrough measurement kernel basis' functions.

${}_{\epsilon}\phi^{(\ell)}_{i^{(\ell)}}$ A collection of linearly independent functions that are used to realize self-reproducing kernels of the integral transforms of measurement due to lateral interactions in the ℓ-th layer of a multi-layer neural signal processor. $i^{(\ell)} = 1, 2, \; . \; {}_{\epsilon}N^{(\ell)}$, ${}_{\epsilon}N^{(\ell)} = 1, 2, . . \;$. These functions are termed 'reproducing lateral measurement kernel basis' functions.

${}_{v}\phi^{(\ell)}_{i^{(\ell)}}$ A collection of linearly independent functions that are used to realize self-reproducing kernels of the integral transforms of aggregation in the ℓ-th layer of a multi-layer neural signal processor. $i^{(\ell)} = 1, 2, \ldots {}_{v}N^{(\ell)}$, ${}_{v}N^{(\ell)} = 1, 2, \ldots$. These functions are termed 'reproducing aggregation kernel basis' functions.

${}_{w}\beta^{(\ell)}_{\imath_1^{(\ell)}\imath_2^{(\ell)}}$ The coefficients of the sum of products of 'reproducing feedthrough measurement kernel basis' functions that are used to synthesize the kernels of the integral transforms of measurement due to feedthrough associations. The values of these coefficients are given by the inverse of the Gramm matrix constructed from the mutual innerproducts of the 'reproducing feedthrough measurement kernel basis' functions.

${}_{\epsilon}\beta^{(\ell)}_{\imath_1^{(\ell)}\imath_2^{(\ell)}}$ The coefficients of the sum of products of 'reproducing lateral measurement kernel basis' functions that are used to synthesize the kernels of the integral transforms of measurement due to lateral interactions. The values of these coefficients are given by the inverse of the Gramm matrix constructed from the mutual innerproducts of the 'reproducing lateral measurement kernel basis' functions.

${}_{v}\beta^{(\ell)}_{\imath_1^{(\ell)}\imath_2^{(\ell)}}$ The coefficients of the sum of products of 'reproducing aggregation kernel basis' functions that are used to synthesize the kernels of the integral transforms of aggregation. The values of these coefficients are given by the inverse of the Gramm matrix constructed from the mutual innerproducts of the 'reproducing aggregation kernel basis' functions.

$\tilde{\mathfrak{X}}$ The collection of concepts (signals) over which an appropriate neural signal processor is defined to realize the 'reproducing kernel basis' functions.

$\tilde{x}$ Concepts (signals) in $\tilde{\mathfrak{X}}$.

${}_{a}\tilde{x}_{\imath^{(\ell)}}$ Directions in the concept (signal) space $\tilde{\mathfrak{X}}$. The directional derivatives, along these directions, of an appropriately chosen neural signal processor are used as the 'reproducing kernel basis' functions.

$\mathfrak{h}$ The neural signal processor (scalar, non-evolutionary and type-1) over $\tilde{\mathfrak{X}}$ whose directional derivatives are chosen as the 'reproducing kernel basis' functions.

$\mathcal{Z}$ A denumerable collection of integers.

${}_w a^{(\ell)}(i^{(\ell)})$ Functions defined to choose the scale factors of the 'reproducing kernel basis' functions in layer ℓ. Other entities of a similar nature are ${}_\epsilon a^{(\ell)}(i^{(\ell)})$ and ${}_v a^{(\ell)}(i^{(\ell)})$. The prefixes '$w$', '$\epsilon$' and '$v$' denote the scaling associated with the basis functions of the kernels of measurement due to feedthrough, measurement due to lateral interaction and aggregation, respectively.

${}_w b^{(\ell)}(i^{(\ell)})$ Functions defined to choose the translations of the 'reproducing kernel basis' functions in layer ℓ Other entities of a similar nature are ${}_\epsilon b^{(\ell)}(i^{(\ell)})$ and ${}_v b^{(\ell)}(i^{(\ell)})$. The prefixes '$w$', '$\epsilon$' and '$v$' denote the shifts associated with the basis functions of the kernels of measurement due to feedthrough, measurement due to lateral interaction and aggregation, respectively.

References

Aazhang, *Behnaam*, **Paris**, *Bernd-Peter* and **Orsak**, *Geoffrey C* (July 1992). Neural networks for multi-user detection in code-division multiple acess communications *IEEE Transactions on Communications*, **40**(7):1212–1222.

Albus, *J S* (1975) A new approach to manipulator control: The cerebellar model articulation controller (CMAC). *Transactions of the American Society of Mechanical Engineers–Journal of Dynamical Systems, Measurements and Control*, **97**:220–227.

Aleksander, *Igor* (1983a). *Artificial Vision for Robots*, New York, USA. Chapman & Hall

Aleksander, *Igor* (1983b). Memory networks for practical vision systems. Design calculations, in *Artificial Vision for Robots* (Edited by **Aleksander**, *Igor*), pages 197-2 14, New York, USA. Chapman & Hall.

Amari, *Shun-Ichi* (September/October 1983). Field theory of self organizing neural nets. *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC13**:741–748

Amit, *Daniel J* (1989). *Modeling Brain Function: The world of attractor neural networks* Cambridge University Press, Cambridge, UK

Anderson, *J A* (September-October 1983). Cognitive and psychological computation with neural models. *IEEE Transactions on Systems, Man, and Cybernetics*, **13**:799–815.

Anderson, *J A*, **Silverstein**, *J W*, *et al.* (1977) Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review*, **84**:413–451

Anderson, *James A* and **Rosenfeld**, *Edward* (1989). *Neurocomputing Foundations of research*. MIT Press, Cambridge, MA, USA.

Antognetti, *Paolo* and **Milutinović**, *Veljko* (1991) *Neural Networks: Concepts, applications, and implementations*, volume I, Englewood Cliffs, New Jersey, USA. Prentice Hall.

Arbib, *Michael* (1975). Artificial intelligence and brain theory: Unities and diversities. *Biomedical Engineering*, **3**·238–274.

Arnol'd, *V I* (1957). On functions of three variables. *Dokl Akad. Nauk. SSSR*, **114**:953–956.

Arnol'd, *V I* (1958). Representation of continuous functions of three variables by the superposition of continuous functions of two variables, in *Translations of the AMS*, pages 61–147.

Aronszajn, *N* (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, **68**·337–404.

Baer, *Jean-Loup* (October 1984). Computer architecture. *Computer*, **17**(10):77–87.

Barnsley, *Michael F* (1988). *Fractals Everywhere*. Academic Press, New York (USA).

Baum, *Eric B* (1986). Towards practical 'neural' computation for combinatorial optimization problems. In **Denker** (1986), pages 53–58. Volume 151.

Baum, *Eric B* and **Haussler**, *D* (1989). What size net gives valid generalization? *Neural Computation*, **1**:151–160.

Ben-Israel, *Adi* and **Greville**, *T N E* (1974) *Generalized Inverses: Theory and applications*. John Wiley, New York, USA.

Benedetto, *John J* (1992). Irregular sampling and frames, in *Wavelets; A tutorial in theory and applications* (Edited by **Chui**, *Charles K*), volume 2 of *Wavelet Analysis and its Applications*, pages 445–507. Academic Press, Inc., New York, USA.

Bertsekas, *Dimitri* and **Gallager**, *Robert* (1987). *Data Networks*. Prentice Hall, Inc., Englewood Cliffs, New Jersey.

Boulding, *Kenneth E* (April 1956). General systems theory–the skeleton of science. *Management Science*, **2**:197–208.

Brink, *Jean R* and **Haden**, *C Roland* (1987). *The Computer and The Brain: Perspectives on Human and Artificial Intelligence*. North-Holland, Amsterdam, The Netherlands © Elsevier Science Publishers.

Bryson, Jr, *Arthur E* and **Ho**, *Yu-Chi* (1969). *Applied Optimal Control: Optimization, Estimation, and Control*. Blaisdell Publishing Company, Waltham, MA, USA.

Burks, *Arthur W* (1970). *Essays on Cellular Automata*. University of Illinois Press, Urbana, USA.

Caianiello, *Eduardo R* (1961). Outline of a theory of thinking machines and thought processes *Journal of Theoretical Biology*, **2**:204.

Carpenter, *Gail* and **Grossberg**, *Stephen* (1986a). Absolutely stable learning of recognition codes by a self-organizing neural network, in *Neural Networks for Computing, American Institute of Physics Conference Proceedings, Vol 151* (Edited by **Denker**, *J*). American Institute of Physics.

Carpenter, *Gail* and **Grossberg**, *Stephen* (1986b). Adaptive resonance theory· Stable self-organization of neural recognition codes in response to arbitrary lists of input patterns, in *Eighth Annual Conference of the Cognitive Science Society*, pages 45–62

Carpenter, *Gail* and **Grossberg**, *Stephen* (1987) ART2: Self-organization of stable category recognition codes for analog input patterns, in *Proceedings of the First International Conference on Neural Networks Vol II* (Edited by **Caudill**, *Maureen* and **Butler**, *Charles*), pages 727–736

Caudill, *Maureen* and **Butler**, *Charles* (1990). *Naturally Intelligent Systems.* The MIT Press, Cambridge, Massachusetts, USA.

Chaturvedi, *Ajit Kumar* (January 1994) *Low Cost and Concurrent Algorithms for Adaptive Volterra and Linear Filters.* PhD thesis, Department of Electrical Engineering, Indian Institute of Technology Kanpur.

Cherry, *Colin* (1957). *On Human Communication· A review, a survey and a criticism.* Studies in Communication. The Technology Press of MIT, Cambridge, Massachusetts, USA Published in conjunction with John Wiley & Sons, Inc., New York, USA

Chui, *Charles K* (1992). *An Introduction to Wavelets.* Academic Press.

Churchland, *Patricia Smith* (1986). *Neurophilosophy.* MIT Press, Cambridge, MA, USA.

Churchland, *Patricia Smith* and **Sejnowski**, *Terence J* (1994). *The Computational Brain.* Computational Neuroscience. MIT Press, Cambridge, Massachusetts. First paperback edition ©1992.

Churchland, *Paul M* (1981). Eliminative materialism and the propositional attitudes. *Journal of Philosophy*, **78** 67–90.

Clementi, *A* and **Mentrasti,** *P* (1990) Cellular automata and neural networks: Links and computational problems, in *Proceedıngs of the Thırd Workshop on Parallel Architectures and Neural Nets, Vıetri (Italy), 1990.*

Cohen, *L* (July 1989). Time frequency distributions–a revıew *Proceedings of the IEEE*, **77**(7):941–981

Cohen, *M A* and **Grossberg,** *Stephen* (September 1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks *IEEE Transactıons of Systems, Man, and Cybernetics*, **13**:815–826.

Cohen, *M A* and **Grossberg,** *Stephen* (December 1987). Masking fields: A massively parallel neural architecture for learning, recognizing, and predicting multiple grouping of patterned data. *Applıed Optıcs*, **26**:1866.

Corwin, *Lawrence J* and **Szczarba,** *Robert H* (1982). *Multivariable Calculus*, volume 64 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel-Dekker, New York.

Cotter, *Neil E* and **Guillerm,** *Thıerry J* (1992). The CMAC and a theorem of kolmogorov. *Neural Networks*, **5**:221–228.

Cover, *Thomas M* (1965) Geometrical and statistical properties of systems of linear inequalities wıth applications in pattern recognition *IEEE Transactıons on Electronıc Computers.*

Cybenko, *George* (1989). Approximation by superpositions of sıgmoidal functions. *Mathematics of Control, Signals, and Systems*, **2**:303–314.

Cybenko, *George* (1990). Complexity theory of neural networks and classification problems, in *Lecture Notes in Computer Scıence* (Edited by **Wellenkens,** *Almeıda*), volume 412. Springer-Verlag, Berlin (Germany).

Dasgupta, *Subrata* (1984). *Computer Archıtecture*, volume 1, Foundatıons. John Wiley and Sons, Inc , New York.

Daubechies, *Ingrid* (1992). *Ten Lectures on Wavelets.* SIAM

Daugmann, *John* (1988). Complete discrete 2-d Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech and Sıgnal Pı ocessıng*, **36** 1169–1179

Davidson, *Jennıfer* and **Hummer**, *Frank* (1993) Morphology neural networks: An introduction with applications *Cırcuıts, Systems, Signal Processing*, **12**(2):177–210.

Denker, *John S* (1986) *Neural Networks for Computıng, AIP Conference Proceedings, Snowbird Utah, 1986*, New York. American Institute of Physics Volume 151.

Dennett, *Danıel C* (1986). The logical geography of computational approaches: A view from the east pole, in *The Representation of Knowledge* (Edited by **Brand**, *M* and **Harnish**, *Mıchael*). The University of Arizona Press, Tucson, AZ, USA

Dennett, *Daniel C* (Winter 1988). When philosophers encounter artificial intelligence. *Daedalus*, **117**(1)·283–295. Published by the Amerıcan Academy of Arts and Sciences.

Derman, *C* (1970) *Finite State Markovıan Decision Process* Academic Press, New York, USA.

Descartes, *Réne* (1960). *Dıscourse on Method (1637).* Bobbs-Merrill, New York, third edition. Translated from the French by Laurence J LaFleur.

Diebold, *John* (1952). *Automation: The advent of the automatic factory.* D van Nostrand Company, Inc, New Jersey.

Dikshit, *H P* and **Micchelli**, *C A* (5–9 January 1993). *International Confeı ence on Advances ın Computational Mathematics*, New Delhi, India. Indira Gandhi National Open University.

Drazin, *P G* (1983). *Solıtons*. Cambridge University Press, Cambridge, England (UK).

Drazin, *P G* and **Johnson**, *R S* (1989) *Solitons: An ıntroduction*. Cambridge University Press, Cambridge, England (UK)

Dreyfus, *Hubert L* and **Dreyfus**, *Stuart E* (Winter 1988). Makıng a mind versus modelıng the brain: Artificial intelligence back at a branch point *Daedalus*, **117**(1)·15–43. Published by the American Academy of Arts and Sciences.

Edelman, *Gerald M* (1987). *Neural Darwinism: The theory of neuronal group selectıon*. Basic Books, Inc

Eden, *Murray* (1983). Cybernetics, in *The Study of Informatıon: Interdısciplınary Messages* (Edited by **Machlup**, *Frıtz* and **Mansfield**, *Una*), pages 409–439. John Wiley & Sons, New York.

Erdös, *Paul*, **Gruber**, *P M* and **Hammer**, *J* (1989). *Lattıce Poınts*. Pitman Monographs and Surveys ın Pure and Applıed Mathematics, No. 39. Longman Scientific and Technical, Essex, UK.

Ershov, *Yu L* and **Palyutin**, *E A* (1984). *Mathematıcal Logıc* MIR Publishers, Moscow. Translated from the Russian by Vladimir Shokurov.

Fahlman, *Scott E* and **Hinton**, *Geoffrey E* (1987). Connectionist architectures for artificial intelligence. *Computer*, **20**:100–109.

Fiesler, *E* (1994). Neural network classification and formalization. Available by anonymous ftp from archive.cis.ohio-state.edu as fiesler.formalızation.ps.Z in the /pub/neuroprose directory.

Fodor, *Jerry A* and **Pylyshyn**, *Zenon W* (1988a) Connectıonism and cognitive architecture: A critical analysis. *Cognitıon*, **28**. Reprinted in **Fodor & Pylyshyn** (1988b).

Fodor, *Jerry A* and **Pylyshyn,** *Zenon W* (1988b). Connectionism and cognitive architecture: A critical analysis, in *Connections and Symbols (A Cognition Special Issue)* (Edited by **Pinker,** *Steven* and **Mehler,** *Jacques*), pages 3–71. MIT Press, Cambridge, MA, USA. Reprinted from **Fodor & Pylyshyn** (1988a).

Fukushima, *Kunihiko* (1969). Visual feature extraction by a multi-layer network of analog threshold elements. *IEEE Transcations on Systems, Man, and Cybernetics*, **5**:322–333.

Fukushima, *Kunihiko* (1970). An electronic model of the retina *Proceedings of the IEEE*, **58**:1950–1951.

Fukushima, *Kunihiko* (1975). Cognitron : A self organizing multi-layered neural network. *Biological Cybernetics*, **20**·121–136.

Fukushima, *Kunihiko* (1980) Neocognitron : A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, **36**.193

Fukushima, *Kunihiko* (December 1987). Neural network model for selective attention in visual pattern recognition and associative recall. *Applied Optics*, **26**(23).4985–4992.

Fukushima, *Kunihiko* and **Miyake,** *Sei* (1982). Neocognitron : A new algorithm for pattern recognition tolerant of deformation and shifts in position. *Pattern Recognition*, **15**:465–475.

Geva, *Shlomo* and **Sitte,** *Joaquin* (1992). A constructive method for multivariate function approximation by multilayer perceptrons. *IEEE Transactions on Neural Networks*, **3**(4):621–624.

Giles, *C L* and **Maxwell,** *T* (December 1987). Learning invariance and generalization in high-order neural networks. *Applied Optics*, **26**:4972–4978

Giles, *C L*, **Griffin**, *R D* and **Maxwell**, *T* (1988). Encoding geometric invariances in higher order neural networks, in *Neural Information Processing Systems* (Edited by **Anderson**, *D Z*), pages 301–309, New York, USA. American Institute of Physics

Girosi, *Federico* and **Poggio**, *Tomaso* (1991). Networks for learning: A view from the theory of approximation of functions, in *Neural Networks: Concepts, applications, and implementations* (Edited by **Antognetti**, *Paolo* and **Milutinović**, *Veljko*), volume I, pages 110–154, Englewood Cliffs, New Jersey, USA. Prentice Hall Chapter 6

Goldberg, *David E* (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesly, Reading, Massachusetts, USA.

Golden, *Richard M* (1988). A unified framework for connectionist systems. *Biological Cybernetics*, **59**(2):109–120.

Gorsline, *George W* (1986). *Computer Organization: Hardware/Software*. Prentice Hall, Inc , Englewood Cliffs, New Jersey.

Grant, *P M* and **Sage**, *J P* (1986) A comparison of neural network and matched filter processing for detecting lines in images. In **Denker** (1986), pages 194–199. Volume 151.

Greene, *Peter H* (1962). On the representation of information by neural net models, in *Self-Organizing Systems* (Edited by **Yovits**, *Marshall C*, **Jacobi**, *George T* and **Goldstein**, *Gordon D*), Washington DC, USA. Spartan Books.

Grossberg, *Stephen* (1980). How does a brain build a cognitive code? *Psychological Review*, **87**:1–51

Grossberg, *Stephen* (1982). *Studies of Mind and Brain · Neural principles of learning, perception development, cognition and motor control* D Reidel, Dordrecht

Grossberg, *Stephen* (1988) Nonlinear neural networks: Principles, mechanisms and architectures. *Neural Networks*, **1**(1):17–61

Haken, *Hermann* (1977). *Synergetics. An introduction* Springer-Verlag, Berlin.

Haken, *Hermann* (1983). *Advanced Synergetics: Instability hierarchies of self-organizing systems and devices.* Springer-Verlag, Berlin.

Hartshorne, *Robin* (1977). *Algebraic Geometry* Springer-Verlag, Berlin.

Hawkins, *J K* (1961). Self-organizing systems – a review and commentary *Proceedings of the IRE*, pages 31–48

Haykin, *Simon* (1984). *Introduction to Adaptive Filters.* MacMillan College Publishing Company, New York, USA

Haykin, *Simon* (1994). *Neural Networks. A comprehensive foundation* MacMillan College Publishing Company, New York, USA

Hazewinkel, *M* (1988). *Encyclopedia of Mathematics: An updated and annotated translation of the Soviet 'Mathematical Encylopedia'.* Kluwer Academic Publishers, Dordrecht, The Netherlands

Hebb, *D O* (1949). *Organization of Behaviour*. John Wiley and Sons, New York (USA)

Hecht-Nielsen, *Robert* (1987a). Counterpropagation networks. *Applied Optics*, **26**:4979–4985. Also presented in **Hecht-Nielsen**, 1987b.

Hecht-Nielsen, *Robert* (1987b). Counterpropagation networks, in *Proceedings of the First International Conference on Neural Networks, Vol II* (Edited by **Caudill**, *Maureen* and **Butler**, *Charles*), pages 113–12, 19–32. Also presented in **Hecht-Nielsen**, 1987a.

Hecht-Nielsen, *Robert* (1987c). Kolmogorov's mapping neural network existence theorem, in *Proceedıngs of the First Internatıonal Conference on Artıficial Neural Networks*, pages III–11–III–14. IEEE Press.

Hecht-Nielsen, *Robert* (1990) *Neurocomputing.* Addison-Wesley Publishing Company, Reading, MA, USA.

Hertz, *John*, **Krogh**, *Anders* and **Palmer**, *Richard G* (1991). *Introduction to the Theory of Neural Computatıon.* Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley, Redwood City, Californıa.

Hinton, *G E* (1989). Deterministic boltzmann learning performs steepest descent in weight-space. *Neural Computatıon*, **1**(1):143–150

Hinton, *Geoffrey E*, **Sejnowski**, *R J* and **Ackley**, *D H* (1984). Boltzmann machınes: Constraint satisfaction networks that learn.

Hodgkin, *A L* and **Huxley**, *A F* (1952). A quantıtative description of membrane current and its application to conduction and excitation in a nerve. *Journal of Physıology*, **117**:500–544.

Hofstadter, *Douglas R* (1979). *Gödel, Escher, Bach: An eternal golden braid.* Penguin, London

Hopcroft, *John E* and **Ullman**, *Jeffery D* (1989) *Introductıon to Automata Theory, Languages, and Computatıon.* Narosa Publıshing House, New Delhi, India.

Hopfield, *John J* (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedıngs of the Natıonal Academy of Scıences, USA*, **79**:2554–2558. Reprinted in **Hopfield**, 1992

Hopfield, *John J* (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedıngs of the National Academy of Scıences (USA)*, **81**:3088–3092.

Hopfield, *John J* (1992). Neural networks and physical systems with emergent collective computational abilities, in *Artificial Neural Networks: Paradigms, applications and hardware implementations* (Edited by **Sánchez-Sinencio**, *Edgar* and **Lau**, *Clifford*), pages 25 29. IEEE Press, New York (USA). Reprinted from **Hopfield**, 1982.

Hopfield, *John J* and **Tank**, *D W* (1985) Neural computation of decision optimization problems. *Biological Cybernetics*, 52.141- 152.

Hornik, *K*, **Stinchcombe**, *Maxwell* and **White**, *Halbert* (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Hoskins, *R F* (1979). *Generalised Functions*. Ellis Horwood Limited, Chichester, West Sussex, England, UK.

Hurst, *S L* (1971) *Threshold Logic* Mills & Boon Limited

Itô, *Kiyoshi* (1987) *Encyclopedic Dictionary of Mathematics*, volume II. Mathematical Society of Japan, second edition

Ivankhnenko, *A G* (October 1971) Polynomial theory of complex systems *IEEE Transactions on Systems, Man, and Cybernetics*, **1**·364–378.

James, *William* (1890). *Psychology (Briefer Course)*, chapter XVI, pages 253–279. Holt, New York, USA. Reprinted in **William James**, 1989.

James, *William* (1989). Psychology (briefer course), in *Neurocomputing: Foundations of Research* (Edited by **Anderson**, *James A* and **Rosenfeld**, *Edward*), chapter 1, pages 4–14. MIT Press, Cambridge, MA, USA. Reprinted from **William James**, 1890.

Jayadeva (June 1993) *Optimization with Neural Networks*. PhD thesis, Department of Electrical Engineering, Indian Institute of Technology, Delhi.

Jeffrey, *W* and **Rosner,** *R* (1986). Neural network processing as a tool for function optimization In **Denker** (1986), pages 241–246. Volume 151.

Jenkins, *W K* (1987). Analog signal processing, in *Encyclopedıa of Physical Scıence and Technology* (Edıted by **Meyers,** *Robert A*), pages 571–598, New York, USA. Academic Press, Inc.

Judd, *J Stephen* (1990). *Neural Network Design and the Complexity of Learnıng.* MIT Press, Cambridge, MA, USA

Kirkpatrick, *S*, **Gelatt,** *C D* and **Vecchi,** *M P* (13 May 1983). *Scıence,* 220:671–680

Kleinrock, *Leonard* (1975). *Queueing Systems.* John Wıley, New York, USA.

Klir, *George J* (1977). *An Approach to General Systems Theory.* van Nostrand Reinhold Company, New York, USA

Kohavi, *Zvi* (1978). *Switching and Fınıte Automata Theory.* Tata McGraw-Hill Publishing Company Ltd., New Delhi, India, second edition.

Kohonen, *Teuvo* (1972). Correlation associative memory. *IEEE Transcatıon on Computers,* **21**:353–359.

Kohonen, *Teuvo* (1977). *Associative Memory: A system theoretıcal approach.* Springer-Verlag, New York, USA.

Kohonen, *Teuvo* (1980). *Content Addressable Memorıes.* Springer-Verlag.

Kohonen, *Teuvo* (1981). Automatic formation of maps in a self-organizıng system, in *Proceedıngs of the Second Scandinavian Conference on Image Analysıs* (Edited by **Oja,** *E* and **Simula,** *O*), pages 214–220

Kohonen, *Teuvo* (1984). *Self Organızation and Associatıve Memory.* Sprınger Serıes in Information Sciences. Springer-Verlag, Berlin

Kohonen, *Teuvo* and **Mäkisara**, *K* (1986) Representation of sensory information in self-organizing feature maps. In **Denker** (1986), pages 271–276. Volume 151.

Kohonen, *Teuvo*, **Mäkisara**, *Kai*, *et al* (1991). *Artificial Neural Networks: Proceedings of the 1991 International Conference on Artificial Neural Networks (ICANN-91), Espoo, Finland, 24–28 June, 1991*, Amsterdam, The Netherlands. North-Holland In 2 Volumes.

Kolmogorov, *A N* (1957a) On the representation of functions of several variable by superpositions of functions of one variable and addition *Translations of the American Mathematical Society*. Translated from the Russian original **Kolmogorov** (1957b).

Kolmogorov, *A N* (1957b) On the representation of functions of several variable by superpositions of functions of one variable. *Dokl Nauk*. In Russian

Kosko, *B* (January 1988). Bidirectional associative memories. *IEEE Transactions on Systems, Man, and Cybernetics*, **18**:49–60

Kosko, *Bart* (1992a) *Neural Networks and Fuzzy Systems: A dynamical systems approach to machine intelligence.* Prentice Hall, Englewood Cliffs, New Jersey, USA.

Kosko, *Bart* (1992b) *Neural Networks for Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, USA.

Kosko, *Bernard* (December 1987). Adaptive bidirectional associative memories. *Applied Optics*, **26**(23):4947–4960.

Kosslyn, *S M* and **Hatfield**, *G* (1984). Representation without symbol systems. *Social Research*, **51**:1019–1054.

Kovačec, *Alexander* and **Ribeiro**, *Bernardete* (1993). Kolmogorov's theorem: From algebraic equations and nomography to neural networks, in *Artificial*

Neural Nets and Genetic Algorithms Proceedings of the International Conference in Innsbruck, Austria, 1993 (Edited by **Albrecht**, *Rudolf F*, **Reeves**, *Colin R* and **Steele**, *Nigel C*), pages 40–47, New York. Springer-Verlag.

Kreyszig, *Erwin* (1978) *Introductory Functional Analysis with Applications*. John Wiley, New York, USA.

Krishnan, *V Sankriti* (1981). *An Introduction to Category Theory*. North-Holland, New York.

Kuhn, *Thomas S* (1962) *The Structure of Scientific Revolutions*. International Encyclopedia of Unified Science, Foundations of the Unity of Science, Vol 2, No 2. The University of Chicago Press, Chicago, USA.

Kůrková, *Věra* (1992). Kolmogorov's theorem and multilayer neural networks. *Neural Networks*, **5**:501–506.

Lagunas, *Miguel A*, **Pérez-Neira**, *Ana*, *et al.* (1992) The Kolmogorov signal processor, in *New Trends in Neural Computation: Proceedings of the International Workshop on Artificial Neural Networks, IWANN'93, Siges, Spain, June 1993* (Edited by **Mira**, *J*, **Cabestany**, *J* and **Prieto**, *A*), volume 686 of *Lecture Notes in Computer Science*, pages 494–512, Berlin, Germany. Springer Verlag.

Lamb, Jr, *George L* (1980). *Elements of Solition Theory*. Pure and Applied Mathematics. John Wiley & Sons, New York, USA.

Lawson, Jr, *H Blaine* (May 1974). Foliations *Bulletin of the American Mathematical Society*, **80**(3):369–418.

Lax, *Peter D* (1968). Integrals of nonlinear equations of evolution and solitary waves. *Communications of Pure and Applied Mathematics*, **21**:467–490.

Lehmann, *E L* (1986) *Testing Statistical Hypotheses*. John Wiley & Sons, New York, USA

Leshno, *Moshe*, **Ya Lin**, *Vladimir*, *et al.* (1994). Multilayer feedforward networks with a non-polynomial activation function can approximate any function. Preprint.

Lewis, *Harry R* and **Papadimitriou**, *Christos H* (1981). *Elements of the Theory of Computation.* Prentice-Hall, Englewood Cliffs, NJ, USA

Lippmann, *Richard P* (April 1987). An introduction to computing with neural nets. *IEEE ASSP Magazine*, pages 4–22.

Lorentz, *G C* (1962) Metric entropy, widths, and superpositions of functions. *American Mathematical Monthly*, **69**:469–485.

Lyon, *Richard F* and **Mead**, *Carver* (July 1988). An analog electronic cochlea. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **36**(7) 1119–1134.

Lyon, *Richard F* and **Mead**, *Carver* (1990). An analog electronic cochlea In **Morgan** (1990), pages 79–94. Reprinted from **Lyon & Mead**, 1988.

Machlup, *Fritz* and **Mansfield**, *Una* (1983). Cultural diversity in studies of information, in *The Study of Information: Interdisciplinary Messages* (Edited by **Machlup**, *Fritz* and **Mansfield**, *Una*), pages 3–56. John Wiley & Sons, New York.

Mandelbrot, *Benoit B* (1987). Fractals, in *Encyclopedia of Physical Science and Technology*, pages 579–593. Academic Press, Inc., New York, USA.

Matheus, *Christopher* and **Hohensee**, *William E* (December 1987). Learning in artificial neural systems. Technical Report Report No. UIUCDCS-R-87-1394, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA

McClelland, *James L*, **Rumelhart**, *David E* and **The PDP research group** (1986a). *Parallel Distributed Processing : Explorations in the microstructure of cognition*, volume 1 MIT Press, Cambridge, Massachusetts.

McClelland, *James L*, **Rumelhart**, *David E* and **The PDP research group** (1986b) *Parallel Distributed Processing : Explorations in the microstructure of cognition*, volume 2. MIT Press, Cambridge, Massachusetts.

McCorduck, *Pamela* (1979). *Machines Who Think.* W H Freeman, San Francisco, CA, USA

McCulloch, *Warren S* and **Pitts**, *Walter* (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5.115–133. Reprinted in **McCulloch & Pitts**, 1965.

McCulloch, *Warren S* and **Pitts**, *Walter* (1965). A logical calculus of the ideas immanent in nervous activity, in *The Embodiments of Mind* (Edited by **McCulloch**, *Warren S*). MIT Press, Cambridge, Massachusetts. Reprinted from **McCulloch & Pitts**, 1943.

Mead, *Carver* (1989). *Analog VLSI and Neural Systems.* Addison-Wesley, Reading, Massachusetts, USA.

Mead, *Carver* and **Ismail**, *M* (1989). *Analog VLSI Implementation of Neural Systems.* Kluwer Academic Press.

Meyers, *Robert A* (1987). *Encyclopedia of Physical Science and Technology*, New York, USA. Academic Press, Inc.

Mhaskar, *H N* (5–9 January 1993). Approximation properties of a multilayered feedforward artificial neural network, in *International Conference on Advances in Computational Mathematics* (Edited by **Dikshit**, *H P* and **Micchelli**, *C A*), New Delhi, India Indira Gandhi National Open University.

Millman, *Jacob* and **Halkias**, *Christos C* (1967) *Electronic Devices and Circuits.* McGraw-Hill, New York.

Minsky, *Marvin Lee* and **Papert**, *Seymour A* (1969) *Perceptrons · An introduction to computational geometry.* MIT Press, Massachusetts. An expanded version of this book is available (*cf* **Minsky** & **Papert**, 1990).

Minsky, *Marvin Lee* and **Papert**, *Seymour A* (1990). *Perceptrons: An introduction to computational geometry.* MIT Press, Massachusetts. In memory of Frank Rosenblatt. This is an expanded version of **Minsky** & **Papert** (1969).

Moon, *F C* (1987) Nonlinear dynamics, in *Encyclopedia of Physical Science and Technology* (Edited by **Meyers**, *Robert A*), pages 91–102, New York, USA Academic Press, Inc.

Morgan, *Nelson* (1990). *Artificial Neural Networks: Electronic Implementations*, IEEE Computer Society Neural Networks Technology Series, Los Alamitos, California, USA. IEEE Computer Society Press.

Murata, *Tadao* (April 1989). Petri nets: Properties, analysis, and applications. *Proceedings of IEEE*, 77·541–580.

Narendra, *Kumpati S* and **Parthasarathy**, *Kannan* (March 1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, **1**(1):4–27

Nashed, *M Zuhair* and **Walter**, *Gilbert G* (1991). General sampling theorems for functions in reproducing kernel Hilbert spaces. *Mathematics of Control, Signals, and Systems*, **4**:363–390.

Newell, *Alan* (1983). Intellectual issues in the history of artificial intelligence, in *The Study of Information: Interdisciplinary Messages* (Edited by **Machlup**, *Fritz* and **Mansfield**, *Una*), pages 187–227. Wiley, New York.

Newell, *Alan* and **Simon**, *Herbert* (1981). Computer science as empirical enquiry: Symbols and search, in *Mind Design* (Edited by **Haugeland**, *John*), page 41. MIT Press, Cambridge, MA, USA. Reprinted from **Newell, Shaw** & **Simon**, 1958.

Newell, *Allen*, **Shaw**, *J C* and **Simon**, *Herbert A* (1958) Elements of a theory of human problem solving. *Psychological Review*, **65**(3):151–166.

Nilsson, *Nils J* (1965). *Learning Machines· Foundations of trainable pattern-classifying systems.* McGraw-Hill Book Company, New York, USA

Olazaran, *Mikel* (1993). A sociological history of the neural network controversy, in *Advances in Computers, Vol 37* (Edited by **Yovits**, *Marshall C*), chapter 5, pages 335–425. Academic Press, Inc , New York.

Orfanidis, *Sophocles J* (1988). *Optimum Signal Processing: An introduction.* McGraw-Hill Book Company, New York, USA, second edition

Pao, *Yoh-Han* (1989). *Adaptive Pattern Recognition and Neural Networks.* Addison-Wesley, Reading, MA, USA.

Partrdige, *Derek* and **Wilks**, *Yorick* (1990) *The Foundations of Artificial Intelligence A sourcebook*, Cambridge. Cambridge University Press

Pati, *Y C* and **Krishnaprasad**, *P S* (January 1993). Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations *IEEE Transactions on Neural Networks*, 4(1):73–85.

Penrose, *R* (1955). A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, **51**:406–413.

Pentland, *Alex Paul* (1986). *From Pixels to Predicates.* Ablex Publishing Corporation, Norwood, New Jersey, USA.

Peretto, *Pierre* (1992). *Introduction to the Modeling of Neural Networks* Cambridge University Press, Cambridge, UK

Pitas, *Ioannis* and **Venetsanopoulos** (1990). *Nonlinear Digital Filters. Principles and applications.* The Kluwer International Series in Engineering and Computer Science, VLSI, Computer Architecture and Digital Signal Processing Kluwer Academic Publishers, Boston, USA.

Poggio, *Tomaso* and **Girosi**, *Federico* (September 1990). Networks for approximation and learning. *Proceedings of the IEEE*, 78(9).1481–1497.

Proakis, *John G* (1989) *Digital Communications.* McGraw-Hill Book Company, New York, USA, second edition.

Pylyshyn, *Zenon W* (1984). *Computation and Cognition : Toward a foundation for cognitive science.* MIT Press, Cambridge, Massachusetts.

Rajaraman, *R* (1982) *Solitons and Instantons.* North-Holland, Amsterdam

Rao, *Yarlagadda* and **Hershey**, *John E* (1987). General signal processing, in *Encyclopedia of Physical Science and Technology* (Edited by **Meyers**, *Robert A*), pages 626–646, New York, USA. Academic Press, Inc

Rosenblatt, *Frank* (1958). The perceptron. A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Rosenblatt, *Frank* (1961). *Principles of Neurodynamics.* Spartan.

Rosenblatt, *Frank* (1962). Strategic approaches to the study of brain models, in *Principles of Self-Organization: Transactions of the University of Illinois Symposium on Self-Organization, Robert Allerton Park, 8 and 9 June, 1961* (Edited by **von Foerster**, *Heinz* and **Zopf**, *George W*), pages 385–402. Pergamon Press, London, UK. International Tracts in Computer Science and Technology and their Application.

Rosen, *Robert* (1985). *Anticipatory Systems· Philosophical, mathematical, and methodological foundations.* International Series in Systems Science and Engineering. Pergamon Press, Oxford, UK.

Rudin, *Walter* (1986). *Real and Complex Analysis.* Tata McGraw-Hill Publishing Company Limited, New Delhi, India, second edition

Rumelhart, *D E*, **Hinton**, *G* and **Williams**, *R* (1986). Learning representations by back-propagating errors. *Nature*, **323**·533–536.

Rumelhart, *David E* and **Norman**, *Donald A* (1981) A comparison of models, in *Parallel Models of Associative Memory* (Edited by **Hinton**, *Geoffrey E* and **Anderson**, *James*). Lawrence Erlbaum Associates, Hillsdale, NJ, USA.

Sage, *Jay P* and **Withers**, *Richard S* (1988) Analog nonvolatile memory for neural network implementations, in *Proceedings of the Electrochemical Society Meeting, October 1988, Symposium on Silicon, Nitride and Silicon Dioxide Thin Isolating Films.* The Electrochemical Society, Inc.

Sage, *Jay P* and **Withers**, *Richard S* (1990) Analog nonvolatile memory for neural network implementations. In **Morgan** (1990), pages 22–33. Reprinted from **Sage & Withers**, 1988.

Saylor, *J* and **Stork**, *D G* (1986). Parallel analog neural networks for tree searching In **Denker** (1986), pages 392–397. Volume 151.

Schetzen, *M* (1980). *The Volterra and Wiener Theories of Nonlinear Systems.* John Wiley & Sons, New York, USA.

Schneider, *W* (1987). Connectionism: Is it a paradigm shift for psychology? *Behavior Research Methods, Instruments, & Computers*, **19**·73–83.

Sejnowski, *Terence J* (1981). Skeleton filters in the brain, in *Parallel Models of Associative Memory* (Edited by **Hinton**, *Geoffrey E* and **Anderson**, *A J*). Erlbaum, Hillsdale, NJ, USA.

Serra, *J* (1982) *Image Analysıs and Mathematical Morphology*, volume 1. Academic Press, London, UK.

Serra, *J* (1988). *Image Analysıs and Mathematıcal Morphology*, volume 2 Theoretical Advances. Academic Press, London, UK.

Shrier, *S*, **Barron**, *R L* and **Gilstrap**, *L O* (1987). Polynomial and neural networks: Analogies and engineering applications, in *IEEE First Internatıonal Conference on Neural Networks*, pages II–431–II–439, San Diego, USA. SOS Printing.

Smolensky, *Paul* (1990). Connectionism and the foundations of AI In **Partridge & Wilks** (1990), pages 306–326

Sompolinsky, *Haım* (1987). The theory of neural networks: The Hebb rule and beyond, in *Proceedings of the Heidelberg Colloquim on Glassy Dynamics and Optimzatıon, June 1986*, pages 1–43, Berlin Springer-Verlag.

Sompolinsky, *Haim* (December 1988) Statistical mechanics of neural networks. *Physics Today*, pages 70–80

Souček, *Branko and the IRIS group* (1992). *Fast Learnıng and Invarıant Object Recognitıon: The sixth generation breakthrough*, New York. John Wiley & Sons

Specht, *D F* (June 1967). Generation of polynomial discriminant functions for pattern recognition. *IEEE Transactions on Electronıc Computers*, **16**:308–319.

Specht, *D F* (April 1967). Vectorcardiographic diagnosis using the polynomial discriminant method of pattern recognition. *IEEE Transactions on Bio-Medıcal Engıneering*, **14**:90–95.

Spirkovska, *Lılly* and **Reid**, *Max B* (1992). Higher order neural networks in position, scale and rotation invariant object recognition. In **Soucek** (1992), pages 153–184.

Sprecher, *David A* (March 1965). On the structure of continuous functions of several variables. *Transactions of the AMS*, pages 340–355.

Starkermann, *Rudolf* (1993) The functional intricacy of neural networks: A mathematical study, in *Artificial Neural Nets and Genetic Algorithms: Proceedings of the International Conference in Innsbruck, Austria, 1993* (Edited by **Albrecht**, *Rudolf F*, **Reeves**, *Colin R* and **Steele**, *Nigel C*), pages 18–24, New York. Springer-Verlag

Stonham, *T J* (1983). Networks of memory elements: A processor for industrial automation, in *Artificial Vision for Robots* (Edited by **Aleksander**, *Igor*), pages 155–178, New York, USA. Chapman & Hall.

Stonier, *Tom* (1990). *Information and the Internal Structure of the Universe: an exploration into information physics*. Springer-Verlag, London.

Tanaka, *Hozumi* (1991). *Artificial Intelligence in the Pacific Rim: Proceedings of the Pacific Rim International Conference on Artificial Intelligence, Nagoya, 1990*, Tokyo, Japan. Japanese Society for Artificial Intelligence, OHMSHA.

Taylor, *Fred J* (1987). Digital signal processing, in *Encyclopedia of Physical Science and Technology* (Edited by **Meyers**, *Robert A*), pages 599–625, New York, USA. Academic Press, Inc.

Tennison, *B R* (1975). *Sheaf Theory*, volume 20 of *London Mathematical Society Lecture Notes Series*. Cambridge University Press, Cambridge, UK.

Toffoli, *T* and **Margolus**, *N* (1987) *Cellular Automata Machines*. MIT Press, Cambridge, MA, USA

Turing, *Alan Matheison* (1950). Computing machinery and intelligence. *Mind*, 59(236).433.

Turing, *Alan Matheison* (1981). Computing machinery and intelligence, in *The Mınd's I* (Edited by **Hofstadter**, *Douglas* and **Dennett**, *Daniel C*), pages 54–67. Basic Books, New York, USA. Reprinted from**Turing**, 1950.

Usui, *S*, **Nakauchi**, *S* and **Nakano**, *M* (1991). Feature extraction of spectral reflectance of munsell color chips by a five-layered neural network, in *Artıficıal Intelligence in the Pacıfic Rım: Proceedıngs of the Pacific Rim International Conference on Artıficial Intellıgence, Nagoya, 1990* (Edited by **Tanaka**, *Hozumı*), pages 522–527, Tokyo, Japan Japanese Society for Artificial Intelligence, OHMSHA.

Valiant, *L G* (November 1984). A theory of the learnable *Communıcatıons of the ACM*, **27**(11):1134–1142.

van Hemmen, *J L* (October 1986). Spin-glass models of a neural network. *Physical Review A*, **34**:3435–3445.

van Hemmen, *J L*, **Grensing**, *D*, *et al* (1988a) Nonlinear neural networks I. General Theory. *Journal of Statistıcal Physıcs*, **50**(1/2):231–257.

van Hemmen, *J L*, **Grensing**, *D*, *et al.* (1988b) Nonlinear neural networks II Information Processing. *Journal of Statistical Physıcs*, **50**(1/2)·259–293

van Loocke, *Phılip R* (1994). *The Dynamics of Concepts: A connectionıst model*, volume 766 of *Lecture Notes in Artıficial Intelligence (Subseries of Lecture Notes in Computer Scıence).* Springer-Verlag, Berlin, Germany.

Vepsäläinen, *Arı M* (1991). Modeling of dynamic systems with expandable neural networks, in *Artificial Neural Networks: Proceedings of the 1991 International Conference on Artificial Neural Networks (ICANN-91), Espoo, Finland, 24–28 June, 1991* (Edited by **Kohonen**, *Teuvo*, **Mäkisara**, *Kai*, *et al.*), pages 37–42, Amsterdam, The Netherlands. North-Holland In 2 Volumes.

von Bertalanffy, *Ludwig* (1968). *General System Theory: Foundations, Development, Applicatıons.* Braziller, New York, USA

von Foerster, *Heinz* and **Zopf**, *George W* (1962). *Princıples of Self Organızatıon*. International Tracts in Computer Science and Technology and Their Application. Pergamon, London, UK. Transactions of the University of Illinois Symposium on Self-Organization, Robert Allerton Park, 8 and 9 June 1961, Sponsored by Information Systems Branch, US Office of Naval Research.

von Neumann, *John* (1959). *Cellular Automata*

von Neumann, *John* (1966) *Theory of Self-Reproducıng Automata*. University of Illinoıs Press, Urbana, USA. Edited and completed by Arthur W Burks.

Walrand, *Jean* (1988). *An Introductıon to Queueıng Networks*. Prentice-Hall, Inc , Englewood Cliffs, New Jersey, USA.

Wechsler, *Harry* (1990). *Computational Visıon*. Computer Science and Scientific Computıng. Academic Press, Inc., Boston, USA.

Widrow, *Bernard* (1959). Adaptive sampled-data systems – a statistical theory of adaptation, in *1959 WESCON Conventıon Record: Part 4*, pages 74–85.

Widrow, *Bernard* and **Hoff**, *M* (1960). Adaptive switching cırcuits, in *1960 WESCON Conventıon Record: Part 4*, pages 96–104.

Widrow, *Bernard* and **Lehr**, *Mıchael A* (September 1990). 30 years of adaptıve neural networks · Perceptron, madaline, and backpropagation. *Proceedings of IEEE*, **78**(9).1415–1442. Reprinted in **Widrow & Lehr** (1992).

Widrow, *Bernard* and **Lehr**, *Michael A* (1992). 30 years of adaptive neural networks. Perceptron, madaline, and backpropagation, in *Artificial Neural Networks: Paradıgms, applications and hardware implementatıons* (Edited by **Sánchez-Sinencio**, *Edgar* and **Lau**, *Clıfford*), pages 82–108. IEEE Press, New York (USA) Reprinted from **Widrow & Lehr** (1990)

Widrow, *Bernard* and **Winter**, *Rodney* (March 1988) Neural nets for adaptive filtering and adaptive pattern recognition. *Computer*, pages 25–39.

Widrow, *Bernard*, **Winter**, *Rodney G* and **Baxter**, *Robert A* (July 1988). Layered neural nets for pattern recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **36**(7):1109–1118.

Wiener, *Norbert* (1948). *Cybernetics, or Control and Communication in the Animal and the Machine* MIT Press, Cambridge, MA, USA.

Wiener, *Norbert* (1958) *Nonlinear Problems in Random Theory*. Wiley, New York, USA. Based on lectures given at the Massachusetts Institute of Technology.

Witkin, *Andrew P* (1986). Scale space filtering, in *From Pixels to Predicates* (Edited by **Pentland**, *Alex Paul*), pages 5–19, Norwood, New Jersey, USA Ablex Publishing Corporation. Chapter 1 of**Pentland**, 1986.

Wolfram, *Stephen* (1986). *Theory and Applications of Cellular Automata: Including selected papers 1983–1986*. Advanced Series on Complex Systems. World Scientific, Singapore

Wonham, *W M* (March 1989) A language-based control theory for discrete-event systems. Lecture presented at IIT Kanpur

Ya Lin, *Vladimir* and **Pinkus**, *Allan* (5–9 January 1993). Fundamentality of ridge functions, in *International Conference on Advances in Computational Mathematics* (Edited by **Dikshit**, *H P* and **Micchelli**, *C A*), New Delhi, India. Indira Gandhi National Open University.

Yin, *Lin*, **Astola**, *Jaako* and **Neuvo**, *Yrjö* (January 1993). Adaptive stack filtering with application to image processing. *IEEE Transactions on Signal Processing*, **41**(1):162–184.

Yin, *Lin*, **Astola**, *Jaako* and **Neuvo**, *Yrjö* (March 1993). A new class of nonlinear filters – neural filters. *IEEE Transactions on Signal Processing*, **41**(3):1201–1222.

Zhang, *Qinghua* and **Benveniste,** *Albert* (1991). Approximation by nonlınear wavelet networks, pages 3417–3420. IEEE.

Zhang, *Qinghua* and **Benveniste,** *Albert* (November 1992). Wavelet networks. *IEEE Transactions on Neural Networks,* 3(6)·889–898.